



# 逻辑与计算机系统设计基础

---

---

# 1 逻辑代数的公理(5)、定理(8)、代入规则、反对偶规则

• 将逻辑函数表达式F中所有的“ $\cdot$ ”变成“ $+$ ”，“ $+$ ”变成“ $\cdot$ ”；“0”变成“1”，“1”变成“0”；原变量变成反变量，反变量变成原变量。保持原函数中运算顺序不变，得到的新函数为原函数的反函数  $\overline{F}$ 。

◆ 若将逻辑函数表达式F中所有的“ $\cdot$ ”变成“ $+$ ”，“ $+$ ”变成“ $\cdot$ ”，“0”变成“1”，“1”变成“0”，并保持原函数中的运算顺序不变，则所得到的新的逻辑表达式称为函数F的对偶式，并记作F’

## 2.逻辑函数的表达形式

1) 与-或” 表达式

2) “或-与” 表达式

3)标准“与 - 或”表达式  $\Sigma m(1,2,4,7)$

最大项、最小项的概念及性质

4)标准“或 - 与”表达式  $= \Pi M(1,5,7)$

5)代数转换法求标准“与-或” 式的一般步骤

反复使用 $X=X(Y+\bar{Y})$  将表达式中所有非最小项的“与项”扩展成最小项;

6)代数转换法求标准“或-与”式的一般步骤

(1)将函数表达式转换成一般“或-与”表达式;

(2)反复用  $A = (A + B)(A + \bar{B})$  把表达式中所有非最大项的“或项”扩展成最大项。

## 2.逻辑函数的表达形式

7)用真值表/卡诺图法求标准“或-与”式的一般步骤

函数F的真值表

A	B	C	F	
0	0	0	0	→ M <sub>0</sub>
0	0	1	0	→ M <sub>1</sub>
0	1	0	1	
0	1	1	0	→ M <sub>3</sub>
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	0	→ M <sub>7</sub>

对F取值为0的输入直接按最大项来写!

$$F = \prod M(0,1,3,7)$$

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

## 2.逻辑函数的表达形式

7)用真值表/卡诺图法求标准“或-与”式的一般步骤

函数F的真值表

A	B	C	F	
0	0	0	0	→ $m_0$
0	0	1	0	→ $m_1$
0	1	0	1	
0	1	1	0	→ $m_3$
1	0	0	1	
1	0	1	1	
1	1	0	1	
1	1	1	0	→ $m_7$

对F取值为0的输入还是按最小项来写！得到F的标准与或式，再去反即可：

$$\bar{F} = \sum m(0,1,3,7)$$

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

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

### 3.逻辑函数化简

代数化简法

关注“或-与”表达式化简的常用方法

- 必要时采用两次对偶法：

- (1)求函数 $F$ 求对偶, 得到“与-或”表达式 $F'$
- (2)求出 $F'$ 的最简“与-或”表达式
- (3)对 $F'$ 再次求对偶,即可得到 $F$ 的最简“或-与”表达式。

如何判断已经达到最简是一个难题！



### 3.逻辑函数化简

样题!

1.完成下列逻辑代数的相关问题

- 1) 用代数法将逻辑函数  $Y1=\bar{A}C+A\bar{C}+B\bar{C}+\bar{B}C$  变换成标准与-或式并化简为最简“与-或”式 (6 分)。
  - 2) 用卡诺图法将上式化简为最简或-与式(3 分)。
  - 3) 比较用卡诺图和代数法化简的优缺点, 并简要从工程角度分析将逻辑函数化简为不同形式最简式的意义。(3 分)
- \*说明: 化简都要求给出详细过程



## 4.数据表示 （第三章）

- 1.机器数表示及其特点
- 2.校验码的特性：无错结论和有错结论是否可信的问题
- 3.CRC校验
- 4.定点数与IEEE 754互换及异常运算结果分析





## 4.数据表示 (第三章)

1)求(-103.5)<sub>10</sub>对应的32位IEEE754格式的浮点数, 给出完整的过程且最后的结果以16进制方式给出。

2.某 C 程序段如下:

```
#include "stdio.h"
#include "conio.h"
main()
{   int z=-1;
    int x = 65534 ;
    int y = 60000;
    printf("\n");
    printf(" z = %x" ,z);
    printf("\n");
    z = x + y;
    printf(" x + y = %d" ,z); }
```

(说明:  $2^{15} = 32768$ ,  $2^{14} = 16384$ ,  $2^{13} = 8192$ ,  $2^{11} = 2048$ ,  $2^9 = 512$ , 依次类推)

完成下列各题:

- 1)若该程序段中第二个 printf 的输出结果为: Z=FFFF, 则该机的机器字长为多少位? 采用的机器数类型是什么? (4 分)
- 2)在相同机器字长的情况下, 上述程序段第四个 printf 的输出结果为: x+y= ( ), 解释该结果并说明产生该现象的原因, 简要说明检测该异常情况的方法。(8 分)
- 3)上述程序段中使用了十六进制和十进制, 计算机内既然采用了二进制, 为什么还要采用十六进制和十进制? (2 分)

## 5.组合逻辑电路设计-第四章

1.组合逻辑电路分析-- 功能与时间特性

2.设计工具、设计方法(含迭代设计)

设计流程

1. 建立给定问题的逻辑描述?

代数法



真值表法

2. 求出逻辑函数的最简表达式



3. 选择器件并对表达式变换



4. 画出逻辑电路图

型号	功能	PDT <sub>MAX</sub>
74LS86	4-2异或	30ns
74LS32	4-2或门	22ns
74LS00	4-2与非门	15ns
74LS04	6-非门	15ns
74LS08	4-2与门	20ns

涉及无关项的问题设计-- 比如 余3码等

3.竞争与险像



## 5.组合逻辑电路设计-第四章

1.硬件迭代设计先设计输入位数相对较少的功能模块，然后通过功能模块的级联构成输入位数可扩展的同功能电路。硬件迭代设计可缓解组合逻辑设计工具的诸多不足。

采用硬件迭代方法设计“扫描法求补”电路。扫描法求补的基本方法如下：

假设二进制数最左的数据位为最高位，对该二进制数从左边向右逐位扫描，直到找到第1个“1”之后，将输入的<sub>2</sub>进制位逐位取反输出，其它情况下直接输出对应的输入位，例如

待求补的二进制数(假定2位一个分组)    **10 10 10 00**

求补后的二进制数(假定2位一个分组)    **01 01 10 00**

1)设计一个2位二进制求补电路模块，该电路的输入如下：

(1)二位二进制数输入端； (2)1位级联输入控制信号**C0**，该位为1时表示前面的扫描中已经找到了第一个“1”；  
(3)1位求补控制端输入端**S**，该位为1时表示要对输入的2位二进制数求操作，至于对该模块的输入二进制数做什么样的处理，则还要配合的**C0**取值；

该电路的输出如下：

(1)二位二进制数输出端； (2)1位级联输出控制信号**C1**，用于多模块级联时，为1时向高位模块内传递在前面的扫描求补中已经找到了第一个“1”；

根据扫描求补举例和关于该模块的输入、输出端说明，用组合逻辑设计方法设计该求补电路模块，给出设计过程，包括真值表、化简和画电路图(8分)；

2)分析该电路是否存在险像？若存在险像请指出是哪种类型的现象？(4分)

3)基于多个上述模块级联(用包含应有输入输出的方框表示上述模块电路，即类似于在**LOGISIM**中对电路进行封装后的模块)构建8位二进数求补电路(3分)；

4)简要分析说明采用迭代方法进行组合逻辑设计的原因(2分)。

## 6. 运算方法与运算器- 第五章

- 1.运算方法：定点数加、减、乘、除；浮点数加、减运算方法(一般表示)
- 2.定点运算溢出检测
- 3.定点运算设计



## 7. 同步时序电路- 第六章

### 1. 常用触发器的功能--激励表

$Q^n \rightarrow Q^{n+1}$	J	K
0     0	0	×
0     1	1	×
1     0	×	1
1     1	×	0

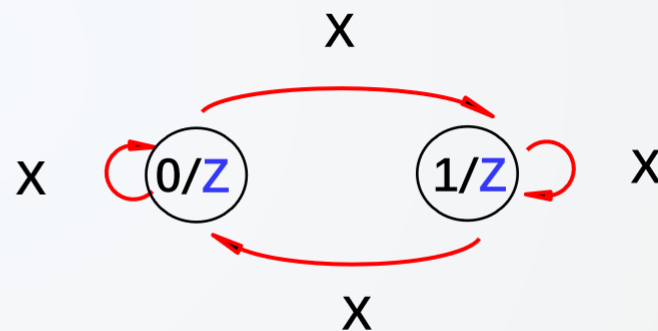
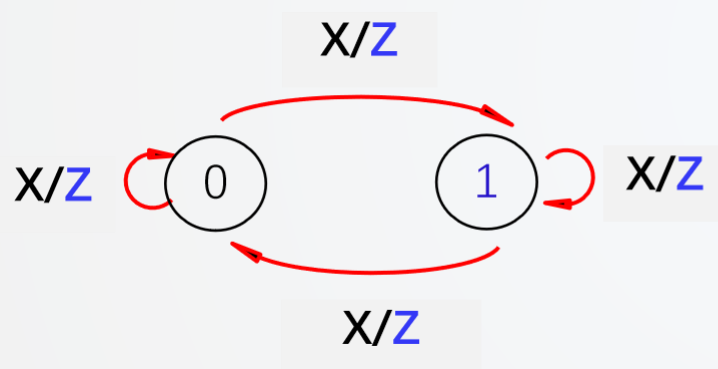
$Q^n \rightarrow Q^{n+1}$	T
0     0	0
0     1	1
1     0	1
1     1	0

$Q^n \rightarrow Q^{n+1}$	D
0     0	0
0     1	1
1     0	0
1     1	1

## 7. 同步时序电路- 第六章

### 2.同步时序电路分析-- 工具、方法

- 1)写出输出函数和激励函数表达式
- 2)写出输出函数和激励函数表达式
- 3)作出状态表和状态图



注意MEALY MOORE型电路的区别

## 7. 同步时序电路- 第六章

### 3.同步时序电路设计-- 工具、方法

1)逻辑抽象，得出电路的状态转换图或状态转换表

(1)进行逻辑定义

(2)状态化简

(3)状态分配及编码

(4)选定触发器的类型设计电路

(5)电路分析检查

▶注意选定的电路类型

▶原始状态图设计是关键

▶重点考查码表状态及**1001**序列检测电路

## 7. 同步时序电路- 第六章

2.设计“1001”序列检测器的同步时序电路MEALY型状态图。(6分)

该电路典型输入和输出对应关系为:

典型输入：0 1 0 0 0 1 0 0 1 0 0 1 0 0 0 1

典型输出：0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0

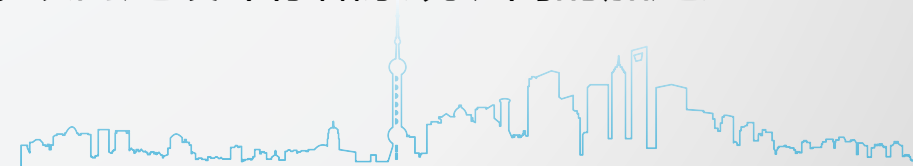


## 8. 存储系统- 第七章

1. 存储系统层次结构及其工作原理-- 局部性原理及其相关例题分析
2. 存储容量与地址线的关系（注意区分SRAM-DRAM）
3. 整数边界存放、大小端存放
4. 存储器扩展--含非连续地址空间的扩展(留有保留区、不同地址区域存储器类型不同等特殊情况)
5. Cache
6. 虚拟存储器

2. 某机内存**16MB**, **CACHE**数据区容量为**16KB**, 每块**8**个字, 每个字**32**位, 采用四路组相联。(15分)

- 1) 根据组相联映射方法, 求该组相联映射中对主存地址划分后各字段位数;
- 2) 设**CACHE**初始状态为空, 若**CPU**顺序访问**0-99**号单元, 并从中读出**100**个字, **CPU**每次读一个字, 并重复此顺序**10**次, 采用**LRU**替换算, 计算**CACHE**命中率?
- 3) 若**CACHE**的速度是主存速度的**6**倍, 求该高速缓冲存储系统访问的加速比。



## 9. 指令系统- 第八章

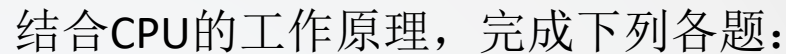
- 1.指令格式与寻址方式 ---尤其是不同寻址方式的特点
- 2.指令格式设计 -- 含操作码字段扩展
- 3.了解RISC的特点

### 4.完成下列关于指令系统的问题。

某计算机字长 16 位，主存 64K，指令采用单字长单地址结构，要求至少能支持 80 条指令和四种寻址方式。

- 1)请设计指令格式并给出指令各字段的位数。(2 分)
- 2)根据不同数据寻址方式的特点、考虑程序员编程方便和指令执行性能等要求，给出你为该计算机设计的四种寻址方式，并分析说明为什么要选择这四种寻址方式？(10 分)

## 1.CPU设计



2.2为什么该CPU不是多周期CPU？如果要将其变成多周期CPU需要在如图所示的哪些位置增加什么部件？(8分)

2.3若该计算机使用的指令有R、J和型3种格式，各字段位数分别如下图所示：

- (1) 该机最多可支持的I型和J型指令总条数为多少？最多可支持的R型指令有多少条？(4分)
- (2) 该机器能使用的寄存器最多有多少个(2分)
- (3) 采用上述指令格式时，任何一条指令能支持几种寻址方式？为什么？(3分)