

# 《数字逻辑与处理器基础》第一次习题课布尔代数/组合逻辑部分

朱振华 zhuzhenh18@mails.tsinghua.edu.cn 03/20/2022

# 目录

- 答疑及习题课说明
- •知识点复习
- 作业给分标准及题目讲解

# 答疑及习题课说明

- 习题课按照内容章节划分,不定周次进行
  - 预计包括:布尔代数与组合逻辑,时序逻辑,指令集与汇编(包括大作业说明及作业题讲解两次),处理器,存储器,期中考试试题讲解
  - 后续习题课前会发放问卷调查同学们倾向的习题课时间, 具体时间待定
  - 习题课主要内容:知识点复习+作业讲解+代表性问题
- •答疑安排
  - 每周日晚7:00~9:00,线上+线下
  - 代表性问题(比如教材中非作业的题目)会在习题课统一讲解

# 知识点复习

- •数的表示与布尔代数
  - •数制转化,格雷码BCD码十进制二进制十六进制
  - •二补码与减法(负数)计算
  - 德摩根定律
  - •逻辑函数对偶与求反
  - 最小项最大项及相互转化
  - •卡诺图化简

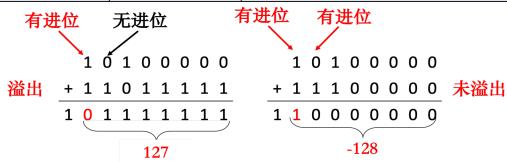
# 易错点:二补码计算与溢出

- 2-补码计算
  - 正数二补码和原码一致
  - 负数二补码计算方式: 负数原码(符号位为1)的非符号位取 反+1
  - •另一种计算方式:2<sup>n</sup>-N, n为总比特数, N为负数绝对值
    - 例3bit -3: 23-3=5=101, 注意最后一步是表示为无符号数的形式

# 易错点: 二补码计算与溢出

- •溢出
  - •溢出判断规则:
    - 直接转为十进制判断是否在表示范围内-2n-1~2n-1-1(n为比特数)
    - 对数字最高位有效位进位和符号位进位做异或,异或结果为1溢出, 异或结果为0不溢出(电路实现:异或门)

符号位进位	最高位进位	结果
0	0	正确
0	1	溢出,上溢(正数相加变负数)
1	0	溢出,下溢(负数相加变正数)
1	1	正确



# 易错点:二补码计算与溢出

- •溢出
  - 第一种溢出情况:符号位无进位,最高位有进位
    - 只有一种情况: 两个正数相加变负数
    - 有一个数是负数,则当最高位有进位时,符号位一定有进位
  - 正数相加结果为负数

	有	进位	<u>ù</u>					
	0	1	0	0	0	0	0	0
+	0	1	0	0	0	0	0	0
	1	0	0	0	0	0	0	0

# 易错点:二补码计算与溢出

- •溢出
  - 第二种溢出情况:符号位有进位,最高位无进位
    - 只有一种情况: 两个负数相加变正数
    - 有一个数是正数,则当符号位有进位时,最高位一定有进位
  - 负数相加结果为正数

有	进	<u>V</u>						
	1	0	0	0	0	0	0	0
+	1	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0

### 易错点: 二补码计算与溢出

- •溢出
  - •溢出判断规则:
    - 对数字最高位有效位进位和符号位进位做异或, 异或结果为1溢出
  - 为什么符号位和最高位同时均有进位时没有溢出?

$$(-64) + (-64) = (-128)$$

绝对值较小负数+绝对值较小负数=负数 绝对值较大的正数+绝对值较小的负数=正数 (64) + (-32) = (32)

### 易错点:对偶及求反的区别

- •1. 形式的区别
- •对偶的形式:与和或互换,逻辑0和1互换,同时保持布尔变量不变得到
  - 例(X+Y+Z+1)D=X·Y·Z·O

- 求反的形式:与和或互换,逻辑0和1互换,同时保持布尔变量求反得到
  - 例(X+Y+Z+1)'=X'·Y'·Z'·O

### ▶ 易错点:对偶及求反的区别

- 2. 用法的区别
- •对偶的用法:验证<mark>布尔代数定理</mark>是否成立,一个布尔代数定理成立,其对偶亦成立
  - $f(\dots) = g(\dots) \Leftrightarrow (f(\dots))^D = (g(\dots))^D$
  - 布尔函数定理,比如重叠定理 X+X=X,对左右两边同时求对偶,可得: X·X=X

• 求反的用法:对布尔函数进行化简或使用特定的门进行映射(比如只使用与非门实现布尔函数)

$$\bullet f = (f')'$$

- •回顾最小项和最大项的定义
  - •最小项:在n变量逻辑函数中,包含全部n个变量的与项, 其中每个变量必须而且只能以原变量或反变量的形式出现 一次,这样的与项称为最小项,用m表示

最小项	使最小项取值为1的输入变量值 最小项下标					
	A	В	C	i		
A'B'C'	0	0	0	0	$m_0$	
A'B'C	0	0	1	1	$\mathbf{m}_1$	
A'BC'	0	1	0	2	$m_2$	
A'BC	0	1	1	3	$m_3$	
AB'C'	1	0	0	4	$m_4$	
AB'C	1	0	1	5	$m_5$	
ABC'	1	1	0	6	$m_6$	
ABC	1	1	1	7	$m_7$	
权重	$2^2$	21	$2^{0}$			

- •回顾最小项和最大项的定义
  - 最大项: 在n变量逻辑函数中, 包含全部n个变量的或项, 其中每个变量必须而且只能以原变量或反变量的形式出现 一次, 这样的或项称为最大项, 用M表示

最大项	使最大项	权值为0的	最大项下标i	$M_{i}$	
	A	В	C		
A+B+C	0	0	0	0	$M_0$
A+B+C'	0	0	1	1	$M_1$
A+B'+C	0	1	0	2	$M_2$
A+B'+C'	0	1	1	3	$M_3$
A'+B+C	1	0	0	4	$M_4$
A'+B+C'	1	0	1	5	$M_5$
A'+B'+C	1	1	0	6	$M_6$
A'+B'+C'	1	1	1	7	$M_7$
权重	$2^2$	21	$2^0$		

- •回顾最小项和最大项的定义
  - 组合逻辑标准式:任意逻辑函数可由最小项的<mark>或</mark>,或者最 大项的与得到
  - SOP: sum of products  $f(A, B, C, D) = AB + A\overline{C} + C\overline{D}$
  - POS: product of sums  $f(A, B, C, D) = (A + B + C)(\overline{C} + \overline{D})$
  - •上面两个式子是最小项的SOP形式和最大项的POS形式吗?

- •回顾最小项和最大项的定义
  - 组合逻辑标准式:任意逻辑函数可由最小项的或,或者最大项的与得到
  - SOP: sum of products  $f(A, B, C, D) = AB + A\overline{C} + C\overline{D}$
  - POS: product of sums  $f(A, B, C, D) = (A + B + C)(\overline{C} + \overline{D})$
  - •上面两个式子是最小项的SOP形式和最大项的POS形式吗?
  - 不是!注意题目要求(用最小项最大项表示,还是用最简 SOP/POS形式表示)

• 最小项和最大项的对应关系

规律: 
$$m_6 = AB\overline{C} = \overline{AB\overline{C}} = \overline{A} + \overline{B} + \overline{C} = \overline{M}_6$$
  
转化示例:  $Y = \sum m(2,3,4,7)$   
 $\overline{Y} = \overline{m_2 + m_3 + m_4 + m_7}$   
 $= m_0 + m_1 + m_5 + m_6$   
 $= \overline{M}_2 + \overline{M}_3 + \overline{M}_4 + \overline{M}_7$   
 $= \overline{M}_2 \bullet \overline{M}_3 \bullet \overline{M}_4 \bullet \overline{M}_7$   
 $= M_2 \cdot M_3 \cdot M_4 \cdot M_7$   
 $= \prod M(2,3,4,7)$   $Y = \prod M(0,1,5,6)$ 

- •最小项和最大项的对应关系
  - 为什么 $Y' = \Pi M(2,3,4,7), Y = \Pi M(0,1,5,6)$
  - 使用卡诺图来理解:

A\BC	00	01	11	10
0	M0	M1	M3	M2
1	M4	M5	M7	M6

注意最大项的下标是该项 取0时确定的

	A/BC	00	01	11	10
Y'	0	1	1	0	0
	1	0	1	0	1

A\BC	00	01	11	10
0	0	0	1	1
1	1	0	1	0

 $\Pi M(2,3,4,7)$ 

 $\Pi M(0,1,5,6)$ 

- •最小项和最大项的对应关系
  - 根据这个例子可知:  $Y = \Sigma m(2,3,4,7), Y' = \Pi M(2,3,4,7), Y = \Pi M(0,1,5,6)$
  - 所以最大项和最小项的下标关系是互补
  - n个变量的全集0~2n-1
    - 3个变量的全集 0~7
    - 4个变量的全集0~15

- •最小项和最大项的对应关系
  - 如何处理无关项呢?
  - 还以上面的例子为例:  $Y = \Sigma m(2,3,4,7) + d(0)$

A\BC	00	01	11	10
0	m0	m1	m3	m2
1	m4	m5	m7	m6

A\BC	00	01	11	10
0	ı	0	1	1
1	1	0	1	0

A\BC	00	01	11	10
0	MO	M1	М3	M2
1	M4	M5	M7	M6

$$Y = \Pi M(1,5,6)D(0)$$

无关项保留,最大项下标去除 最小项下标和无关项下标

- •卡诺图的画法:一定要注意格雷码编码顺序
  - 为什么?
  - 相邻两项之间只有一个输入是不一样,从而可以进行消元

AB	00	01	11	10
CD	ľ			
00	0	1	1	0
01	1	1	1	0
11 ]	1	0	1	1
10	0	0	1	1

AC,消除了BD变量

- •卡诺图化简代表性错误
  - •错误1: 圈没有画到最大,忽略左右上下连通

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

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

错误画法 A'C'D'+A'C

正确画法 A'D'+A'C

- •卡诺图化简代表性错误
  - •错误2: 没有利用无关项让圈最大
    - 无关项:可以是0 可以是1

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

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

错误画法 AC'D+A'CD

正确画法 AD+CD

- •卡诺图化简代表性错误
  - •错误3:额外圈了不必要的无关项
    - 无关项:可以是0 可以是1

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

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

错误画法 B'C+CD

正确画法 B'C

- 最大项卡诺图的化简方法
  - •最小项:下标是表达式为1的取值; m(i)+m(j), 一个为1则结果为1  $\rightarrow$  圈1,写成SOP
  - 最大项:下标是表达式为0的取值;M(i)M(j),一个为0则结果为0→圈0,写成POS

ВС	00	01	11	10
Α				
0	0	0	0	1
1	0	0 (	1	1
	F=A	B+B	C'	

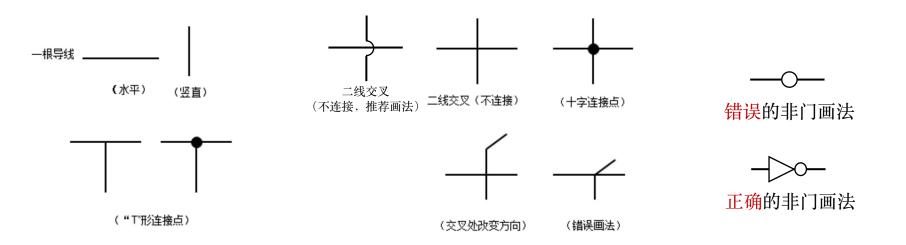
ВС	00	01	11	10	
Α					
0	0 (	0	0	1	
1	0	0	1	1	
F=B(A+C')					

### 知识点复习

- 组合逻辑部分
  - 组合电路功能分析与设计
  - 组合电路冒险与解决方法
  - 噪声容限的定义
  - 门电路转换时间和延时的定义及区别
  - 电路低功耗技术
  - 编码器与译码器
  - 多路选择器的设计与应用
  - 一位全加器、行波进位加法器、超前进位加法器

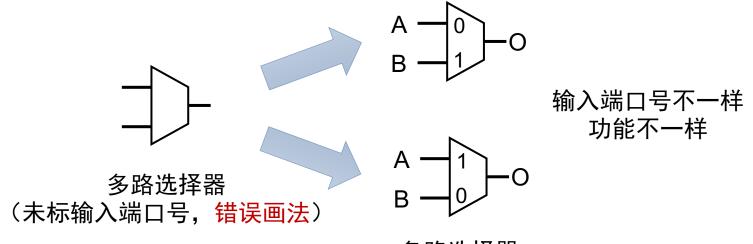
#### 组合电路设计

- •注意电路图设计规范(考试扣分点)
  - 电路元件连接线:应水平或竖直取向,除使用斜线改善易读性的情况外。
  - 不能使用曲线连接, 导线不能穿过电路模块



# 组合电路设计

- •注意电路图设计规范(考试扣分点)
  - 编码器、译码器、多路选择器等电路模块**必须**要写出输入 输出端口号
  - 以多路选择器为例,其他电路模块类似

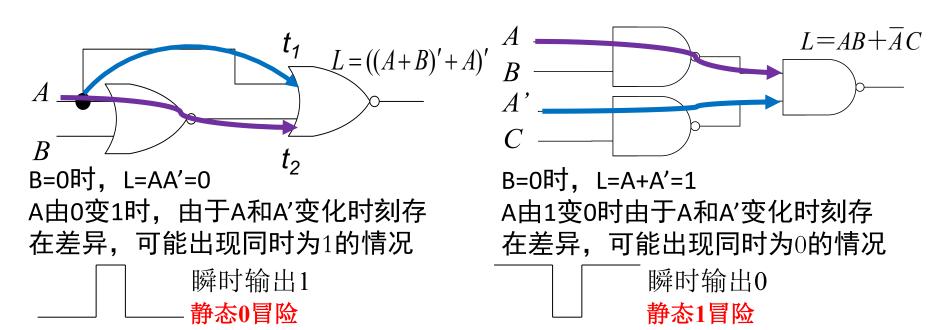


多路选择器 (标出输入端口号,正确画法)

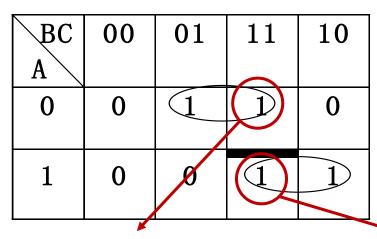
# 组合电路设计

- •设计思路:
  - 从功能出发,人工分析电路应完成的功能,将复杂问题化简为多个基础的子任务
    - 判断5个数中有多少个1 → 将5个数加起来
  - 充分考虑各种情况,避免遗漏
    - 特殊情况的考虑: 条件判断+使用多路选择器选择结果
  - •卡诺图化简不是"万能"的
    - •尽管可以设计出电路,但是设计复杂度过高
    - 不会考察5变量卡诺图化简
  - 自定义模块要说清功能并给出电路实现(标出所有输入输出端口)

- 毛刺:本应保持不变的输出值出现了瞬时变化
- •冒险:有可能产生毛刺的电路存在冒险
  - 冒险是电路的<mark>固有特性</mark>,存在冒险的电路是否会产生毛刺取决于输入值和电路的电特性(比如工作电压等)



- 如果输入初始值和输入最终值能被同一个本原蕴含项覆盖,则不会出现毛刺
  - •原因: 当输入在同一个本原蕴含项内变化时,发生反相的变量与输出无关(例子中的A:1→0)



$$L = AB + \overline{A}C$$

ABC: 111→011, L存在冒险

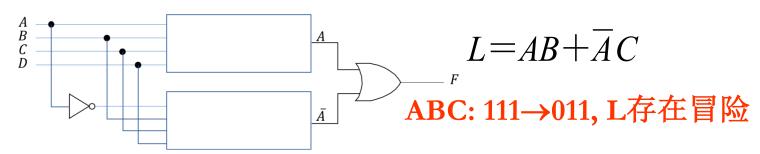
输入最终值011

输入初始值111

• 若电路输出端为或门,且<u>存在其他变量取特定值</u>的情况下表达式化为 $F = A + \bar{A}$ 

• 其他变量取特定值,且A翻转时, $\overline{A}$ 和A不同时变化:导致静态1





- 同理,输出端为与门,存在 $F = A\bar{A}$ 时
  - 静态0冒险

- •消除冒险
  - 将输入初始值和最终值使用同一个本原蕴涵项覆盖,则不会出现毛刺

BC	00	01	11	10	
A			输入	最终值:	
0	0	1	1)4	0	
1	0	0	1	1	
			<u>†</u>		
<del></del>					
输入初始值: 111					

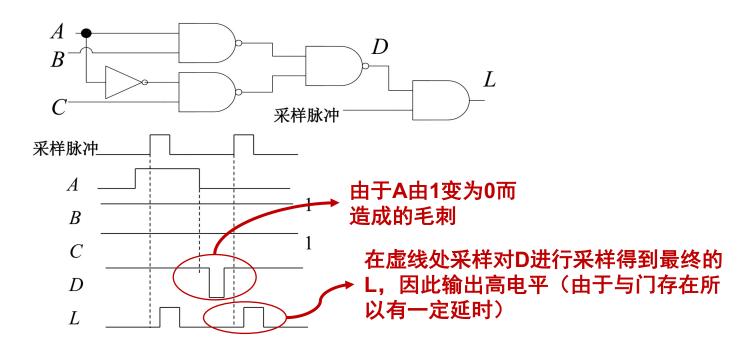
此时卡诺图表示函数:  $F = AB + \bar{A}C + BC$  当ABC由111变为011时,此时即使A  $\bar{A}$ 均为0,但是由于 011 BC项,输出依然为1,并未产生毛刺

这样操作的原因: 当输入初始值和最终值用一个本原蕴涵项覆盖后, 发生变化的输入变量不会对输出产生影响。在这个例子中, 红色框标识的本原蕴涵项覆盖后, 输入BC为11时, A变量的变化不会对函数输出值产生影响, 因此不会出现静态1冒险

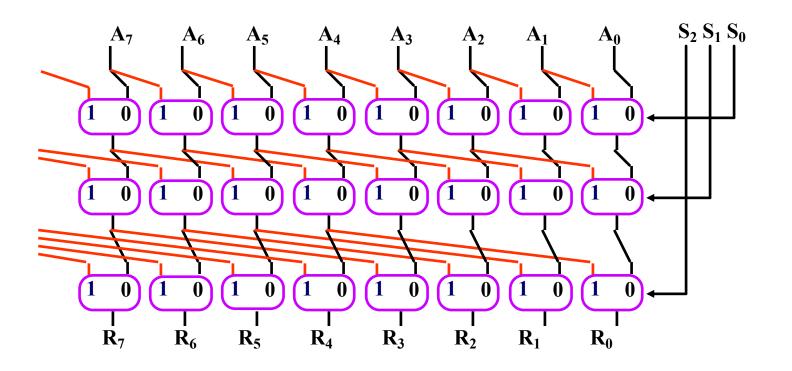
静态0冒险如何判断?利用圈0得到的最大项卡诺图 两个变量同时发生变化的冒险如何消除?无法通过卡诺图化简消除,本课程不涉及

#### •消除冒险

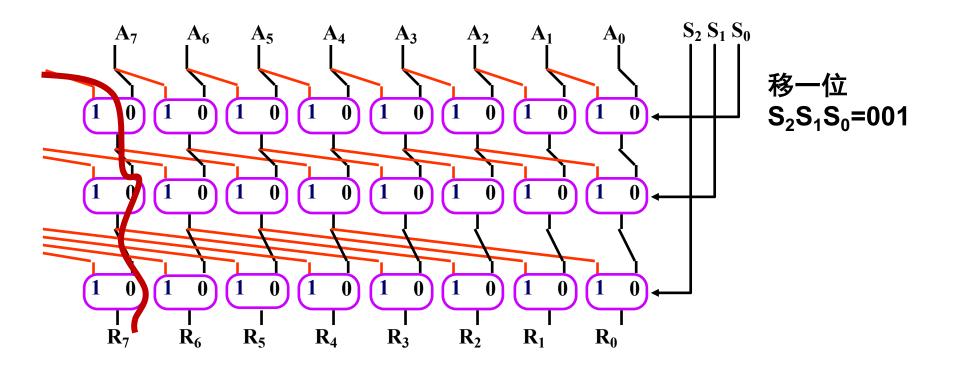
也可以利用采样脉冲进行消除,由于毛刺仅发生在输入信号变化的瞬间, 因此可以在输入信号稳定之后加入采样脉冲,只有采样脉冲作用期间的 输出信号才被视作有效输出信号,则可以避免产生的毛刺影响波形。



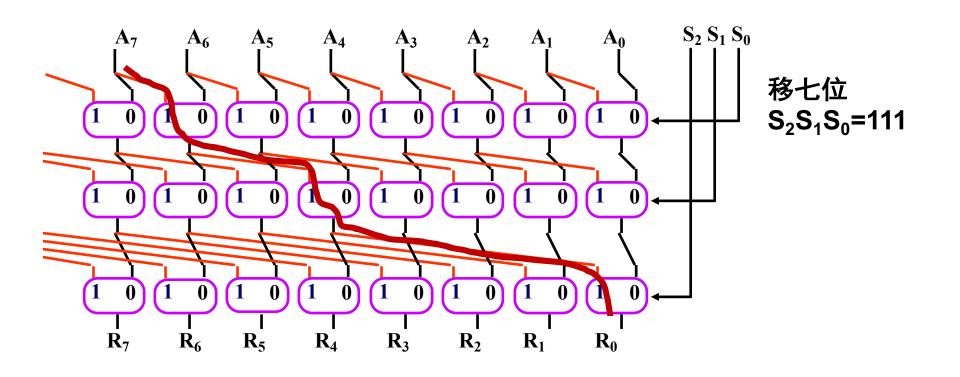
# 多路选择器:基于多路选择器的移位器



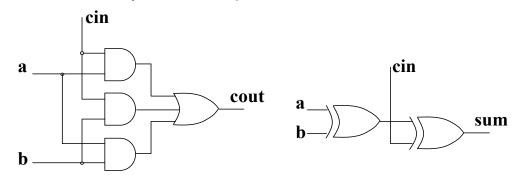
# 多路选择器:基于多路选择器的移位器



# 多路选择器:基于多路选择器的移位器



•一位全加器电路, Cout/sum依赖于Cin:



- $C_{i+1} = A_i C_i + B_i C_i + A_i B_i$
- $C_2 = A_1C_1 + B_1C_1 + A_1B_1$
- $C_1 = A_0C_0 + B_0C_0 + A_0B_0$
- •超前进位加法器的核心思想:将 $C_{i-1}$ 表达式逐级代入到 $C_i$ 中,使 $C_i$ 只与 $C_0$ 有关
  - $C_2 = A_1 \cdot A_0 \cdot B_0 + A_1 \cdot A_0 \cdot C_0 + A_1 \cdot B_1 \cdot C_0 + B_1 \cdot A_0 \cdot B_0 + B_1 \cdot A_0 \cdot C_0 + B_1 \cdot A_0 \cdot C_0 + B_1 \cdot A_1$

- 超前进位加法器如何实现并行:
  - 串行加法器无法并行的原因:每一级 $C_{out}$ 的计算都要利用  $C_{in}$ (即上一级的 $C_{out}$ ),因此需要等待上一级计算完成在计算下一级
  - •理解超前进位加法器的主要原理:将 $C_{out}$ 表示为 $\{A_iB_i\}C_0$ 信号的逻辑函数,直接用加数和被加数计算每一级的进位,而不依赖于前一级的 $C_{in}$ ,因此各级可以并行计算

- •直接写出的超前进位表达式过于复杂:引入P与G
  - P进位传播信号:A与B相加无进位,但是可以把上一级的进位信号传递给下一级: $P = A \oplus B$

А	В	Cin	Cout
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

- •直接写出的超前进位表达式过于复杂:引入P与G
  - P进位传播信号:A与B相加无进位,但是可以把上一级的进位信号传递给下一级: $P = A \oplus B$
  - G进位产生信号:A与B相加本身即可产生进位,与Cin无关:G = AB

Α	В	Cin	Cout
0	0	1	0
0	1	1	1
1	0	1	1
1	1	1	1

- •直接写出的超前进位表达式过于复杂:引入P与G
  - P进位传播信号:  $P = A \oplus B$
  - G进位产生信号: : G = AB
- •借助PG化简迭代式

#### C<sub>1</sub>C<sub>2</sub>C<sub>3</sub>C<sub>4</sub>只与P<sub>i</sub> G<sub>i</sub> C<sub>0</sub>有关,与前一级 Cout无关

$$\begin{split} &C_{i+1} = G_i + P_i C_i = G_i + P_i (G_{i-1} + P_{i-1} C_{i-1}) \\ &= G_i + P_i G_{i-1} + P_i P_{i-1} (G_{i-2} + P_{i-2} C_{i-2}) \\ &\vdots \\ &= G_i + P_i G_{i-1} + P_i P_{i-1} G_{i-2} + \dots + P_i P_{i-1} P_{i-2} \dots P_1 G_0 + P_i P_{i-1} P_{i-2} \dots P_0 C_0 \\ &C_1 = G_0 + P_0 C_0 \\ &C_2 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0 \\ &C_3 = G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0) = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \\ &C_4 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0 \end{split}$$

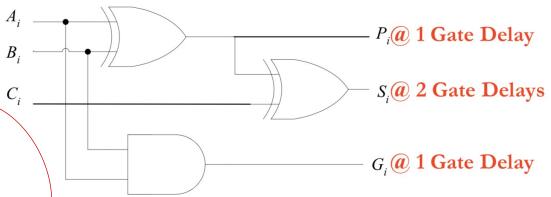
- 超前进位加法器门延时的计算:
  - 主要看经过了几个门  $C_{i+1} = P_i C_i + G_i$

$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

$$S_i = P_i \oplus C_i$$

圆圈标注出来的门 各对应一个门延时



• 以课件上4bit的超前进位加法器为例

$$C_{1} = G_{0} + P_{0}C_{0}$$

$$C_{2} = G_{1} + P_{1}(G_{0} + P_{0}C_{0}) = G_{1} + P_{1}G_{0} + P_{1}P_{0}C_{0}$$

$$C_{3} = G_{2} + P_{2}(G_{1} + P_{1}G_{0} + P_{1}P_{0}C_{0}) = G_{2} + P_{2}G_{1} + P_{2}P_{1}G_{0} + P_{2}P_{1}P_{0}C_{0}$$

$$C_{4} = G_{3} + P_{3}G_{2} + P_{3}P_{2}G_{1} + P_{3}P_{2}P_{1}G_{0} + P_{3}P_{2}P_{1}P_{0}C_{0}$$

◆C<sub>1</sub>/C<sub>2</sub>/C<sub>3</sub>/C<sub>4</sub> 由Pi/Gi两级逻辑产生,相比Pi/Gi有两级 门延时

P<sub>i</sub>/G<sub>i</sub>由一级门延时产生

 $C_1/C_2/C_3/C_4$  在三级门延迟之后产生

 $C_1/C_2/C_3/C_4$  @ 3 Gate Delay

### 目录

- 答疑及习题课说明
- •知识点复习
- 作业给分标准及题目讲解

# 第一次作业(合计10分)

- 答案都在课件里,严禁上网查找摘抄
- 作业有变动,不要照抄往年作业
- 1. 集成电路发展(3分)你认为制约我国集成电路发展的瓶颈有哪些?针对这些瓶颈或困难我国集成电路产业应做哪些努力?
  - 第一问(2分)应该包括如下几方面(但不限于如下几方面): 1. 在集成电路制备的材料、设备等方面受制于人; 2.大陆集成电路制备水平落后于台湾、韩国等地区国家; 3.我国的EDA工具整体水平落后于美国公司,产品不齐全,市场份额小。 4. 已有框架体系受制于现有知识产权体系,颠覆性体系存在诸多技术和行业生态挑战; 5. 整体从业人员数量不足; 6. 长期缺乏市场反哺作用。
  - 第二问(1分)言之有理即可

## 第一次作业

#### • 2. 摩尔定律(4分)

- 摩尔定律具体指什么?通过特征尺寸scaling down可以获得哪些增益,原因是什么?近些年摩尔定律难以继续维持的原因是什么?
- 第一问(1分)同一颗集成电路芯片上集成的器件数量呈指数增长趋势, 增长速度由1975年的每年提高1倍逐渐放缓至每18个月(甚至更长时间) 提高1倍。
- 第二问(1分)器件面积更小,集成度更高,批量制造的制作成本更低(0.5分);器件电容变小,速度更快功耗更低(0.5分)。
- 第三问(2分)工艺尺寸进一步降低会导致芯片功率密度持续增加的问题, 受芯片功耗的物理限制; 晶体管尺寸变小后量子效应显著器件物理特性发生变化; 设计制造成本越来越高(先进工艺在产量不够时无法摊薄)。上述三点少一点扣0.5分

## 第一次作业

- 3. 硬件思路软件思路(3分)
  - •如何理解CPU和ASIC的通用性和性能的权衡,分别具有哪些优势或劣势。从通用性和专用性的角度考虑,你认为未来集成电路的发展趋势是什么?
  - 第一问(2分) CPU的优势:通用性强,处理能力不限于特定算法,使用软件思路解决实际问题,可以很容易地部署多种多样的软件算法;劣势:处理能力和能量效率弱于针对特定算法进行优化的ASIC
  - ASIC的优势:针对特定算法设计开发,使用硬件思路实现,可以在该算法上达到优于CPU的处理速度及能量效率;劣势:只适用于特定的某类算法,通用性差,开发周期长。
  - 第二问(1分)言之有理即可,比如采用两者结合的思路: 面向应用的ASIC+具备通用处理能力CPU的异构计算

## 第二次作业(合计30分,归一化至10分)

- •1. 进制转换(5分)
  - 错一个扣0.5分,没有在最高位补0不扣分
  - •细心即可,严谨地话需要补足对齐

十进制	二进制	十六进制	8421BCD码
124	<u>0111 1100</u>	<u>0x7C</u>	<u>0001 0010 0100</u>
<u>59</u>	0011 1011	<u>0x3B</u>	<u>0101 1001</u>
<u>177</u>	<u>1011 0001</u>	0xB1	0001 0111 0111
<u>39</u>	<u>0010 0111</u>	<u>0x27</u>	0011 1001

- 2.用 8 位二进制数表示十进制数-34<sub>10</sub>、59<sub>10</sub>的原码、 反码和补码
  - 合计5分,每个1分
  - -34:
    - •原码 1010 0010;
    - 反码 1101 1101;
    - 补码 1101 1110;
  - 59
    - 原码=反码=补码 0011 1011

- 3. 请使用卡诺图化简布尔函数  $F(A, B, C, D) = \Sigma m(0,4,5,7,8,12,13,14,15)$ ,要求将结果表示为或与表达式形式(POS, Product of Sums)
  - •卡诺图2分,卡诺图中画错一项扣0.5分。
  - 结果3分,每项1分,写成SOP形式(AB+BD+C'D')扣2分, 结果不是最简扣1分,错误不得分

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

- 3. 请使用卡诺图化简布尔函数  $F(A, B, C, D) = \Sigma m(0,4,5,7,8,12,13,14,15)$ ,要求将结果表示为或与表达式形式(POS, Product of Sums)
  - •方法1: 求F', 卡诺图化简后利用德摩根定律

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

			-	
AB CD	00	01	11	10
00	0	1	1	
01	0	0	0	1)
11	0	0	0	0
10	0	1	1	1

$$F' = B'D + B'C + A'CD'$$

$$F = (B'D + B'C + A'CD')'$$

$$= (B + D')(B + C')$$

$$(A + C' + D)$$

- 3. 请使用卡诺图化简布尔函数  $F(A, B, C, D) = \Sigma m(0,4,5,7,8,12,13,14,15)$ ,要求将结果表示为或与表达式形式(POS, Product of Sums)
  - •方法2:直接圈0写成POS

AB CD	00	01	11	10
00	1	8	9	
01	1	1	1	0
11	1	1	1	
10	1	0	0	0

$$F = (B + D')(B + C')(A + C' + D)$$

- 4.分别用代数化简法和卡诺图化简法化简逻辑函数:  $F = A^{'}B^{'}C^{'} + A^{'}B^{'}C + A^{'}BC + ABC + A^{'}BC'$ 
  - 代数化简法2分, 步骤1分结果1分

• 
$$F = A'B'C' + A'B'C + A'BC + ABC + A'BC'$$
  
=  $A'(B'C' + B'C + BC + BC') + (A + A')BC'$   
=  $A' + BC$ 

• 4.分别用代数化简法和卡诺图化简法化简逻辑函数:  $F = A^{'}B^{'}C^{'} + A^{'}B^{'}C + A^{'}BC + ABC + A^{'}BC'$ 

•卡诺图化简法3分,卡诺图2分,错一项扣0.5;结果1分

C AB	00	01	11	10		
0	1	1	0	0		
1	1	1	1	0		
A' + BC						

- •5.已知函数:  $f(A,B,C) = \sum m(0,1,2,4,5)$ :
  - •1) 将该函数转化为最大项表示的Product-Of-Sums(POS)形式;
  - 2分,写错不得分
  - 3变量下标范围0~7, 直接写出:  $f(A, B, C) = \Pi M(3,6,7)$

- •5.已知函数:  $f(A,B,C) = \sum m(0,1,2,4,5)$ :
  - 2)使用卡诺图将该函数化简为最简Sum-Of-Products(SOP) 形式;
  - 3分,卡诺图及化简过程2分,化简结果1分

C AB	00	01	11	10
0		1	0	
1	1	0	0	1
		D/	11.01	

- •5.已知函数:  $f(A,B,C) = \sum m(0,1,2,4,5)$ :
  - 3)使用下表提供的逻辑门,如果要以最小面积实现该函数的计算功能应如何设计电路,此时面积是多少?如果考虑最小延时呢?(注:门名称后的数字代表输入数;电路输入信号仅包括A,B,C)(5分)
  - •面积最小错误扣2分,延时最小错误扣2分,逻辑正确计算错误扣1分,单位错误扣0.5分

逻辑门	延时(ps)	面积(μm²)	逻辑门	延时(ps)	面积(µm²)
NOT	18	8	AND4	90	40
AND2	50	25	NAND4	70	30
NAND2	30	15	OR4	100	42
OR2	55	26	NOR4	80	32
NOR2	35	16			-

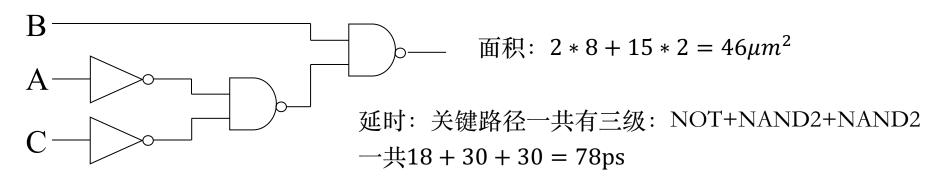
- 5.已知函数:  $f(A,B,C) = \sum m(0,1,2,4,5) = B' + A'C'$ :
- 3) 如何以最小面积/延时完成该函数的计算

逻辑门	延时(ps)	面积(μm²)	逻辑门	延时(ps)	面积(μm²)
NOT	18	8	AND4	90	40
AND2	50	25	NAND4	70	30
NAND2	30	15	OR4	100	42
OR2	55	26	NOR4	80	32
NOR2	35	16			

- •思路: NAND2与NOR2面积延时均小于AND2 OR2
- 应尽可能使用NAND2 NOR2

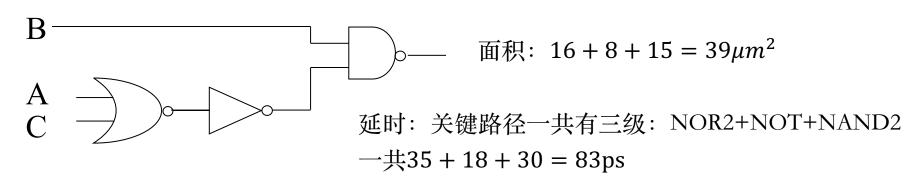
• 
$$f(A,B,C) = B' + A'C' = \overline{\overline{(B'+A'C')}} = \overline{(B(A'C')')} = \overline{(B(A+C))}$$

•  $f(A,B,C) = B' + A'C' = \overline{(B' + A'C')} = \overline{(B(A'C')')} = \overline{(B(A+C))}$ 



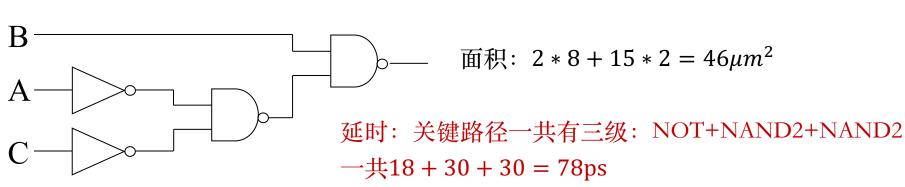
逻辑门	延时(ps)	面积(μm²)	逻辑门	延时(ps)	面积(μm²)
NOT	18	8	AND4	90	40
AND2	50	25	NAND4	70	30
NAND2	30	15	OR4	100	42
OR2	55	26	NOR4	80	32
NOR2	35	16			1

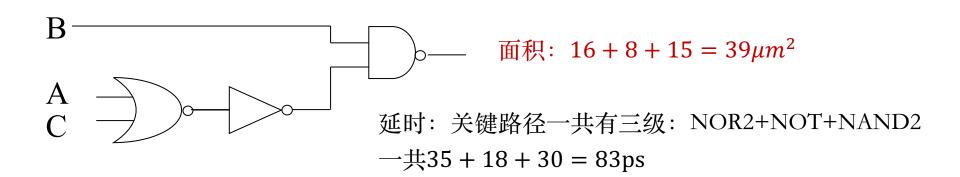
• 
$$f(A,B,C) = B' + A'C' = \overline{(B' + A'C')} = \overline{(B(A'C')')} = \overline{(B(A + C))}$$



逻辑门	延时(ps)	面积(μm²)	逻辑门	延时(ps)	面积(μm²)
NOT	18	8	AND4	90	40
AND2	50	25	NAND4	70	30
NAND2	30	15	OR4	100	42
OR2	55	26	NOR4	80	32
NOR2	35	16			1

• 
$$f(A,B,C) = B' + A'C' = \overline{\overline{(B'+A'C')}} = \overline{\overline{(B(A'C')')}} = \overline{\overline{(B(A+C))}}$$





### 第三次作业(合计10分)

- 1. 根据下列布尔函数的定义,写出化简过程及最简两级与或表达式; 并利用或非门(输入端口数不限)和非门实现电路;判断电路是否存在 冒险,若存在,实现其无冒险电路。
  - a.  $f(A, B, C, D) = \sum m(0,1,3,4,9,11) + \sum d(7,15)$
  - b.  $f(A, B, C, D) = \sum m(0,1,4,5,7,13,15) + \sum d(2,3)$
  - c.  $f(A, B, C, D) = \sum m(1,3,5,7,9) + \sum d(4,11)$

每问0.5分,布尔函数化简0.1分,或非门和非门实现0.2分,电路冒险及判断0.2分

### 第三次作业(合计10分)

 1. 根据下列布尔函数的定义,写出化简过程及最简两级与或表达式; 并利用或非门(输入端口数不限)和非门实现电路;判断电路是否存在 冒险,若存在,实现其无冒险电路。

a. 
$$f(A, B, C, D) = \sum m(0,1,3,4,9,11) + \sum d(7,15)$$

b. 
$$f(A, B, C, D) = \sum m(0,1,4,5,7,13,15) + \sum d(2,3)$$

c. 
$$f(A, B, C, D) = \sum m(1,3,5,7,9) + \sum d(4,11)$$

a.

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

$$f = B'D + A'C'D' = (B + D')' + (A + C + D)'$$
$$= \left( \left( (B + D')' + (A + C + D)' \right)' \right)'$$

存在冒险,当A=B=C=0时, $f = (D')' + D' \equiv 1$  但是当D'从1变换到0时,(D')'和D'存在同时为0的时刻**静态1冒险!** 

消除: 
$$f = B'D + A'C'D' + A'B'C'$$
  
=  $\left( \left( (B + D')' + (A + C + D)' + (A + B + C)' \right)' \right)'$ 

1. 根据下列布尔函数的定义,写出化简过程及最简两级与或表达式; 并利用或非门(输入端口数不限)和非门实现电路;判断电路是否存在 冒险,若存在,实现其无冒险电路。

a. 
$$f(A, B, C, D) = \sum m(0,1,3,4,9,11) + \sum d(7,15)$$

b. 
$$f(A, B, C, D) = \sum m(0,1,4,5,7,13,15) + \sum d(2,3)$$

c. 
$$f(A, B, C, D) = \sum m(1,3,5,7,9) + \sum d(4,11)$$

b.

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

$$f = A'C' + BD = (A + C)' + (B' + D')'$$
$$= \left( \left( (A + C)' + (B' + D')' \right)' \right)'$$

不存在冒险

(注意: 无关项不需要覆盖)

1. 根据下列布尔函数的定义,写出化简过程及最简两级与或表达式; 并利用或非门(输入端口数不限)和非门实现电路;判断电路是否存在 冒险,若存在,实现其无冒险电路。

a. 
$$f(A, B, C, D) = \sum m(0,1,3,4,9,11) + \sum d(7,15)$$

b. 
$$f(A, B, C, D) = \sum m(0,1,4,5,7,13,15) + \sum d(2,3)$$

c. 
$$f(A, B, C, D) = \sum m(1,3,5,7,9) + \sum d(4,11)$$

C.

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

$$f = A'D + B'D = (A + D')' + (B + D')'$$
$$= \left( \left( (A + D')' + (B + D')' \right)' \right)'$$

不存在冒险

- 2. 用一个4:1多路选择器和其它任意逻辑单元,但附加的逻辑要尽可能最少,来实现函数:
  - $f(A,B,C,D,E) = A + \overline{C}D + B\overline{D} + \overline{B}D + \overline{B}CE$
  - 应当使用哪两个信号来控制多路选择器?利用由下列输入信号组合控制的多路选择器实现上述函数。哪一种结果需要最少的逻辑?为什么?
- a. 将A和B作为4:1多路选择器的控制输入
- b. 将B和C作为4:1多路选择器的控制输入
- c. 将B和D作为4:1多路选择器的控制输入
- d. 将C和D作为4:1多路选择器的控制输入
- 1.5分,给出分析过程1分,结果0.5分答案不唯一,言之有理即可

• 2. 用一个4:1多路选择器和其它任意逻辑单元,但附加的逻辑要尽可能最少,来实现函数:

• 
$$f(A,B,C,D,E) = A + \overline{C}D + B\overline{D} + \overline{B}D + \overline{B}CE$$

a. A和B作为控制:

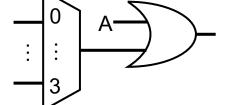
AB	f f
00	C'D+D+CE (= D+CE)
01	C'D+D' (= C'+D')
10	1
11	1

化简前: 2个非门, 2个2输入与门, 3个2输入或门(或者1个3输入或门 1个2输入或门)

化简后: 2个非门, 1个2输入与门, 2个2输入或门

b. B和C作为控制:

ВС	f
00	A+D
01	A+D+E
10	1
11	A+D'



无需进一步化简

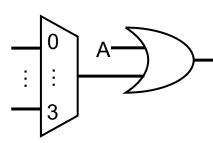
1个非门,3个二输入或门。如果把A移到多路选择器的后面,仅需要1个非门,2个2输入或门

• 2. 用一个4:1多路选择器和其它任意逻辑单元,但附加的逻辑要尽可能最少,来实现函数:

• 
$$f(A,B,C,D,E) = A + \overline{C}D + B\overline{D} + \overline{B}D + \overline{B}CE$$

c. B和D作为控制:

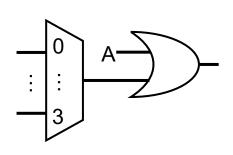
BD	f
00	A+CE
01	1
10	1
11	A+C'



无需进一步化简.1 个非门, 1 个 2 输入与门, 2 个 2 输入或门 A 移到多路选择器后面,则需要 1 个非门, 1 个2输入与门,1个2输入或门

d. C和D作为控制:

CD	f
00	A+B
01	1
10	A+B+B'E = A+B+E
11	A+B'+B'E (=A+B')



化简前:1个非门,1个2输入与门,4个2输入或门;A移到多路选择器后,1非门,1与门,3个2输入或门 化简后:1个非门,3个2输入或门;A移到多路选择器后,1个非门,2个2输入或门

- 3.设计一种优先编码器如图2所示, $I_3I_2I_1I_0$ 为4 bit有权输入, $O_1O_0$ 为编码输出,同 时该编码器将同时输出结果有效位V: V = 1则表示输出 $O_1O_0$ 有效,输出 $O_1O_0$ 为输入  $I_3I_2I_1I_0$  中权重最大的1出现的位置(例如,输入1111时输出11,输入0001时输出00); V = 0则表示输出 $O_1O_0$ 无效,此时输入 $I_3I_2I_1I_0$ 均为 $O_0$ 。请画出该编码器的卡诺图,并使 用与非门和非门实现该电路功能。
- 1.5分,  $VO_1O_0$  每个各0.5分,卡诺图0.2分,化简0.1分,电路实现0.2分

						3   <sub>2</sub>   <sub>1</sub>		4-2优 编码	光 器	$-V$ $-O_1$ $O_0$
.10	00	01	11	10	<b>O</b> 0:	1110		01	11	10
	_					1212	UU	OI	++	10

01:

1110 1312	00	01	11	10
00	*	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$O1 = I2 + I3$$
  
 $O1 = (I2'I3')'$ 

1110  312	00	01	11	10			
00	*	0	1	1			
01	0	0	0	0			
11	1	1	1	1			
10	1	1	1	1			
(	01 =  3 +  1 2'						

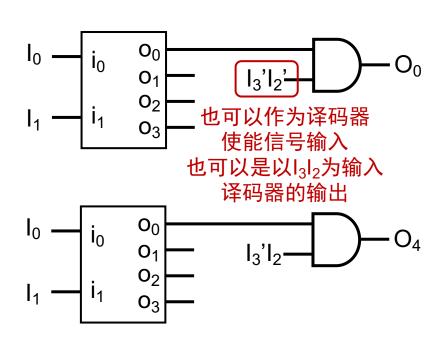
O0 = (13'(1112')')'

<b>:</b>	11 0  3 2	00	01	11	10
	00	0	1	1	1
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

$$V = 11 + 12 + 13 + 14$$
  
 $V = (11'12'13'14')'$ 

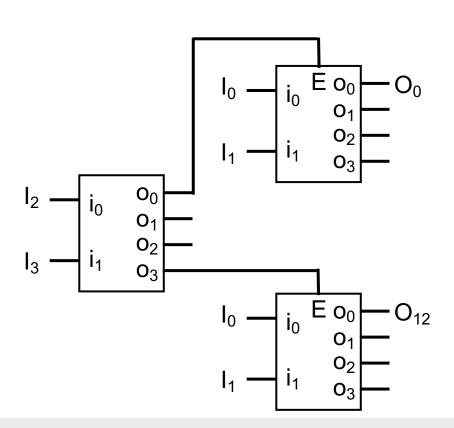
- 4. 使用多个 2:4 译码器实现一个 4:16 译码器(无需考虑使能信号端)
- 1.5分, 答案不唯一,设计功能正确即可,电路模块需要标注端口序号
- •参考设计1:
  - 进行分组,以前8个输出为例

13	12	11	10	00	01	02	О3	04	O5	O6	07
0	0	0	0	1							
0	0	0	1		1						
0	0	1	0			1					
0	0	1	1				1				
0	1	0	0					1			
0	1	0	1						1		
0	1	1	0							1	
0	1	1	1								1



- 4. 使用多个 2:4 译码器实现一个 4:16 译码器(无需考虑使能信号端)
- 1.5分, 答案不唯一,设计功能正确即可,电路模块需要标注端口序号
- •参考设计2:
  - 进行分组,以前8个输出为例

13	12	11	10	00	01	02	О3	04	O5	O6	07
0	0	0	0	1							
0	0	0	1		1						
0	0	1	0			1					
0	0	1	1				1				
0	1	0	0					1			
0	1	0	1						1		
0	1	1	0							1	
0	1	1	1								1



- 5.类比加法器设计减法器:
- 计算A-B的一位全减器与一位全加器类似,其包含两个数据输入(1bit 被减数A与减数B),一个低位的借位输入BL<sub>in</sub>,一个高位的借位请求输出 BL<sub>out</sub>和一个 1bit 差结果输出D。请列出一位全减器的真值表,并给出具体的电路实现;
- 真值表1分, 化简正确0.6分, 电路正确0.4分
- 易错点1: 真值表有误

		1	1	
	1	1	0	0
-	0	0	0	1
	1	0	1	1

A         B         BLin         D         BLout           0         0         0         0         0           0         0         1         1         1           0         1         0         1         1					
0         0         0         0           0         0         1         1         1           0         1         0         1         1	Α	В	BLin	D	BLout
0         0         1         1         1           0         1         0         1         1	0	0	0	0	0
0 1 0 1 1	0	0	1	1	1
	0	1	0	1	1
0 1 1 0 1	0	1	1	0	1
1 0 0 1 0	1	0	0	1	0
1 0 1 0 0	1	0	1	0	0
1 1 0 0 0	1	1	0	0	0
1 1 1 1	1	1	1	1	1

$$D$$

$$= BL_{in}A'B' + BL_{in}AB$$

$$+ BL_{in}AB' + BL_{in}A'B$$

$$= BL_{in} \oplus A \oplus B$$

$$BL_{out}$$

$$= A'B + BL_{in}A' + BL_{in}B$$

- 5.类比加法器设计减法器:
- 类比超前进位加法器实现超前进位减法器,要求写出Pi Gi,并利用Pi Gi BL<sub>0</sub>写出各级Di与BLi;
- Pi Gi各0.5分, Di合计0.5分, BLi合计0.5分
- · 易错点2: Pi表达式有误,未按照要求写出Di与BLi

#### 变换进位的布尔表达式

$$C_{i+1} = B_i C_i + A_i B_i + A_i C_i$$
 $C_{i+1} = \overline{A_i} B_i C_i + A_i \overline{B_i} C_i + A_i B_i \overline{C_i} + A_i B_i C_i$ 
 $= (\overline{A_i} B_i + A_i \overline{B_i}) C_i + A_i B_i (\overline{C_i} + C_i)$ 
 $= P_i C_i + G_i$ 
 $P_i = A_i \oplus B_i$ 
 $G_i = A_i B_i$ 
 $\oplus C_i$ 
 $\oplus C_i$ 

 $S_i = P_i \oplus C_i$ 

Α	В	BLin	D	BLout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

什么时候会产生新的借位: A=0 B=1, A-B有借位产生  $G_i = A_i'B_i$  什么时候A-B无借位,但是会把借位传播到下一级: A=0 B=0, A=1 B=1时  $P_i = \overline{A_i \oplus B_i} = A_i' \oplus B_i$ 

- 5.类比加法器设计减法器:
- 类比超前进位加法器实现超前进位减法器,要求写出Pi Gi,并利用Pi Gi BL<sub>0</sub>写出各级Di与BLi;
- Pi Gi各0.5分, Di合计0.5分, BLi合计0.5分
- · 易错点2: Pi表达式有误,未按照要求写出Di与BLi

$$G_{i} = A'_{i}B_{i} \qquad P_{i} = \overline{A_{i} \oplus B_{i}} = A'_{i} \oplus B_{i}$$

$$D = BL_{in}A'B' + BL_{in}AB + BL_{in}AB' + BL_{in}A'B$$

$$= BL_{in} \oplus A \oplus B = BL_{in} \oplus P'$$

$$BL_{out} = A'B + BL_{in}A' + BL_{in}B$$

$$= (BL + BL_{in})A'B + BL_{in}A'(B + B') + BL_{in}(A + A')B$$

$$= A'B + \overline{A_{i} \oplus B_{i}} BL_{in} = G + PBL_{in}$$

具体递推公式略

## ▶ 第三次作业

- 5. 类比加法器设计减法器:
- 附加题:设计模式可调的超前进位/借位加减法器;
- 易错点: 实现的加减法器不是超前进位/借位
- 思路:比较超前进位加法器与超前借位减法器Pi Gi的关系与区别,类比超前进位加法器进行实现

$$P_{i} = \begin{cases} A_{i} \oplus B_{i}, & Add \\ \overline{A_{i} \oplus B_{i}} = A'_{i} \oplus B_{i}, & Sub \end{cases}$$

$$C_{i+1} = G_{i} + P_{i}C_{i} \qquad BL_{i+1} = G_{i} + P_{i}BL_{i}$$

$$S_{i} = C_{i} \oplus P_{i} \qquad D_{i} = BL_{i} \oplus P_{i}'$$

$$A'_{i}B_{i}, & Sub$$

## ▶ 第三次作业

- 5.类比加法器设计减法器:
- 附加题:设计模式可调的超前进位/借位加减法器;
- 易错点: 实现的加减法器不是超前进位/借位
- 思路:比较超前进位加法器与超前借位减法器Pi Gi的关系与区别,类比超前进位加法器进行实现

$$P_{i} = \begin{cases} A_{i} \oplus B_{i}, & Add \\ \overline{A_{i} \oplus B_{i}} = A'_{i} \oplus B_{i}, & Sub \end{cases}$$

$$C_{i+1} = G_{i} + P_{i}C_{i} \qquad BL_{i+1} = G_{i} + P_{i}BL_{i}$$

$$S_{i} = C_{i} \oplus P_{i} \qquad D_{i} = BL_{i} \oplus P_{i}'$$
对比可知,主要差别在求反上

## ▶ 第三次作业

- 5.类比加法器设计减法器:
- 附加题:设计模式可调的超前进位/借位加减法器;
- 易错点: 实现的加减法器不是超前进位/借位
- 思路:比较超前进位加法器与超前借位减法器Pi Gi的关系与区别,类比超前进位加法器进行实现

$$P_{i} = \begin{cases} A_{i} \oplus B_{i}, Add & C_{i+1} = G_{i} + P_{i}C_{i} & \text{利用M信号实现可控的求反:} \\ \overline{A_{i} \oplus B_{i}} = A'_{i} \oplus B_{i}, Sub & S_{i} = C_{i} \oplus P_{i} & A^{+} = A \oplus M \end{cases}$$

$$G_{i} = \begin{cases} A_{i}B_{i}, & Add & BL_{i+1} = G_{i} + P_{i}BL_{i} \\ A'_{i}B_{i}, & Sub & D_{i} = BL_{i} \oplus P_{i}' \end{cases}$$

#### 谢谢大家

朱振华

问题反馈: zhuzhenh18@mails.tsinghua.edu.cn 03/20/2022