《计算机组织与结构》2020 年第 1 次机考考试说明

学号:_____ 姓名:_____

考试要求

1）请在上方填写学号和姓名，在考试过程中始终出示个人的学生证或一卡通；
2）不得携带与课程相关的任何纸质、电子材料，不需要携带草稿纸；
3）不得携带 U 盘等移动存储设备；
4）考试过程中不得使用手机等通讯设备或出现其它违反考试纪律的行为。

考试说明

1）考试成绩根据完成时间加权：第 1 个小时内完成，权重为 1.1；第 1 至第 2 个小时内完成，权重为 1.0；第 2 至第 3 个小时内完成，权重为 0.9；第 3 至第 4 个小时内完成，权重为 0.8；
2）超过 4 个小时未能完成考试的，本次上机考试成绩为 0；
3）出现考试违纪情况的，课程成绩为 0，并提请教务处处理；
4）考试平台在评测时会隐藏标准输出和标准出错信息。

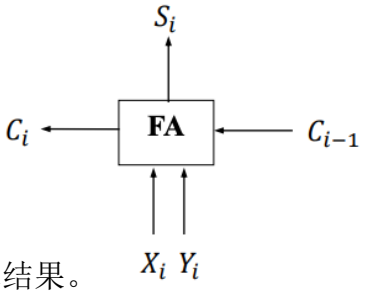
考试题目

1. 实现将整数转化为 16 位补码

方法：String intToComplement(int num)

输入：十进制整数（数值大小不会超过 16 位补码的可表示范围）。
输出：16 位二进制补码。

2. 实现 1 位的全加器



方法：String fullAdder(char x, char y, char c)

输入：x：1 位的二进制数。
　　y：1 位的二进制数。
　　c：进位输入，1 位的二进制数。
输出：长度为 2 的字符串，包括 1 位的进位输入和 1 位的加法运算结果。

3. 实现 4 位的先行进位加法器

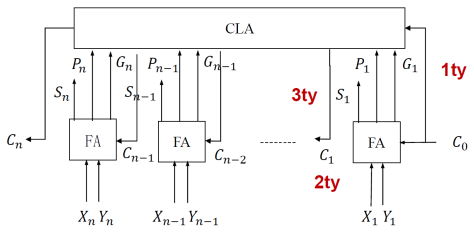
$$C_i = X_iC_{i-1} + Y_iC_{i-1} + X_iY_i$$
$$C_1 = X_1Y_1 + (X_1 + Y_1)C_0$$
$$C_2 = X_2Y_2 + (X_2 + Y_2)X_1Y_1 + (X_2 + Y_2)(X_1 + Y_1)C_0$$
$$C_3 = X_3Y_3 + (X_3 + Y_3)X_2Y_2 + (X_3 + Y_3)(X_2 + Y_2)X_1Y_1 + (X_3 + Y_3)(X_2 + Y_2)(X_1 + Y_1)C_0$$
$$C_4 = \text{.....}$$

Let $P_i = X_i + Y_i \quad G_i = X_iY_i$

$$C_1 = G_1 + P_1C_0$$
$$C_2 = G_2 + P_2G_1 + P_2P_1C_0$$
$$C_3 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1C_0$$
$$C_4 = \text{.....}$$

Carry Look Ahead Adder (cont.)

- Drawback: complex



Latency = 1ty + 2ty + 3ty = 6ty

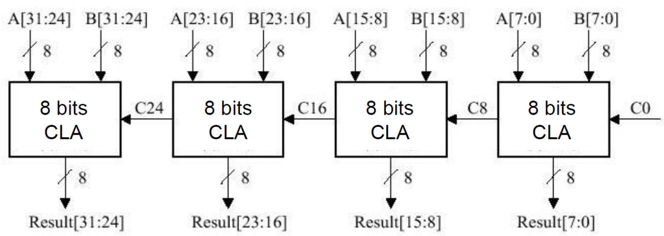
方法：String claAdder(String operand1, String operand2, char c)

输入：operand1：4 位补码。
　　operand2：4 位补码。
　　c：进位输入，1 位的二进制数。
输出：5 位的加法运算结果，包括 1 位的进位和 4 位的和。

4. 实现 16 位的部分先行进位加法器（要求：基于上述方法 claAdder）

Partial Carry Look Ahead Adder

- Idea
  - Serially connect some CLA adders
- Example



Latency = 3ty + 2ty + 2ty + 5ty = 12ty

方法：String pclaAdder(String operand1, String operand2, char c)

输入：operand1: 16 位补码。

operand2: 16 位补码。

c: 进位输入，1 位的二进制数。

输出：17 位的加法运算结果，包括 1 位的进位和 16 位的和。

5. 实现 16 位的布斯乘法

Booth's algorithm

$$X \times Y = X \times Y_n Y_{n-1} \ldots Y_2 Y_1$$

$$= X \times (-Y_n \times 2^{n-1} + Y_{n-1} \times 2^{n-2} + \cdots + Y_2 \times 2^1 + Y_1 \times 2^0)$$

$$= X \times \begin{pmatrix} -Y_n \times 2^{n-1} + Y_{n-1} \times (2^{n-1} - 2^{n-2}) + \cdots \\ +Y_2 \times (2^2 - 2^1) + Y_1 \times (2^1 - 2^0) \end{pmatrix}$$

$$= X \times \begin{pmatrix} (Y_{n-1} - Y_n) \times 2^{n-1} + (Y_{n-2} - Y_{n-1}) \times 2^{n-2} + \cdots \\ +(Y_1 - Y_2) \times 2^1 + (Y_0 - Y_1) \times 2^0 \end{pmatrix} \quad Y_0 = 0$$

$$= 2^n \times \sum_{i=0}^{n-1} \left(X \times (Y_i - Y_{i+1}) \times 2^{-(n-i)} \right)$$

$$P_{i+1} = 2^{-1} \times \left(P_i + X \times (Y_i - Y_{i+1}) \right)$$

Booth's algorithm (cont.)

-7

× -6

42

[X]c = 1001

[-X]c = 0111

[Y]c = 1010

		product	Y	
initial		0000	10100	
Y0-Y1 = 0	->	0000	01010	
Y1-Y2 = -1	-X	0111	01010	
	->	0011	10101	
Y2-Y3 = 1	+X	1100	10101	
	->	1110	01010	
Y3-Y4 = -1	-X	0101	01010	
	->	0010	10101	42

方法：ArrayList<String> multiplication(String operand1, String operand2)

输入：operand1: 被乘数，十进制整数，不会超过 16 位补码的可表示范围。

operand2: 乘数，十进制整数，不会超过 16 位补码的可表示范围。

输出：ArrayList 的长度为 17，第 1 个元素为初始的 product+Y（含 Y0，共 33 位），第 2-17 个元素为计算过程中每次右移后的 product+Y（共 33 位）。