

## 书面作业一

刘泓尊 2018011446 计 84

1. ×

虽然 RISC 的硬件实现相对简单，但是处理器频率也不是必然比 CISC 高，还取决于材料和工艺等。

2. √

加法器可以实现加法和减法，模拟人计算的过程即可实现乘除法，也可以由软件来实现乘除。

3. √

恢复余数法需要先恢复余数（加上除数），余数左移一位，下次再减去除数；但是加减交替法只需要余数左移，下次加上除数。所以加减交替法不需要恢复余数，实现更简单。

4. B

指令的地址来自于程序计数器，数据的地址来自于地址形成部件。

5. D

处理器频率和硬件的复杂程度有关，但是芯片面积和复杂程度没有必然联系，可能芯片上很多器件最终是用不到的。

6. BD

B 是因为  $-2^{30}$ （如果编码为 32 位宽）的原码和补码相同；D:  $y = \text{INT\_MIN}$  即可推出反例。

7. B

UTF-8 使用 1~4 字节为每个字符编码，英文为 1 字节，中日韩文一般为 3 字节。

8. A

B 因程序而异；CD 其实是相对的，硬件操作的速度因指令而异，有些复杂操作 CISC 可以利用硬件实现，但是 RISC 可能需要较多的简单操作。

9.

推导布斯算法：

$x, y$  的实际值与补码表示如下：

若  $[x]_{\text{补}} = x_{n-1}x_{n-2} \cdots x_1x_0$ ，则：

$$x = -2^{n-1}x_{n-1} + \sum_{i=0}^{n-2} x_i 2^i$$

同理：  $[y]_{\text{补}} = y_{n-1}y_{n-2} \cdots y_1y_0$ ，则：

$$y = -2^{n-1}y_{n-1} + \sum_{i=0}^{n-2} y_i 2^i$$

而补码的乘积与乘积的补码相同，所以有

$$\begin{aligned} [x * y]_{\text{补}} &= [x]_{\text{补}} * (-2^{n-1}y_{n-1} + \sum_{i=0}^{n-2} y_i 2^i) \\ &= [x]_{\text{补}} * [2^{n-1}(y_{n-2} - y_{n-1}) + 2^{n-2}(y_{n-3} - y_{n-2}) + \cdots + 2^0(y_{-1} - y_0)] \\ &= [x]_{\text{补}} * \sum_{i=0}^{n-1} 2^i (y_{i-1} - y_i), \text{ 其中 } y_{-1} = 0 \end{aligned}$$

所以可以直接通过 $[y]_{\text{补}}$ 相邻位的差为 1, -1, 0 来决定 $+[x]_{\text{补}}$ 还是 $-[x]_{\text{补}}$ 。而  $2^i$  可以通过对结果移位来实现。所以在最坏情况下，布斯乘法需要  $n$  次加减,  $n$  次移位。需要使用  $n$  位被乘数寄存器,  $n$  位 ALU 和  $2n$  位部分积寄存器，再加初值为 0 的 1 位附加线路。

### 计算 $3 * (-7)$

补码表示:

3: 00011

-3: 11101

-7: 11001

步骤	操作	部分积 $y_i$	$y_{i-1}$	$y_{i-1} - y_i$
0	初值	00000 11001	0	-1
	-3	11101 11001		
1	右移	11110 11100	1	1
	+3	00001 11100		
2	右移	00000 11110	0	0
3	右移	00000 01111	0	-1
	-3	11101 01111		
4	右移	11110 10111	1	0
5	右移	11111 01011	---	---

所以最终结果是 11111 01011(补码) = -21  
和预期结果一致。

### 10.

解下列方程:

$$k = 4$$

$$2^{r-1} \geq k + r$$

得到最小的  $r = 4$

对编码进行排列:

p1, p2, d1, p3, d2, d3, d4, p4

1 2 3 4 5 6 7 8

得到组合:

d1: p1, p2

d2: p1, p3

d3: p2, p3

d4: p1, p2, p3

所以得到海明码表示

$$p_1 = d_1 \oplus d_2 \oplus d_4$$

$$p_2 = d_1 \oplus d_3 \oplus d_4$$

$$p_3 = d_2 \oplus d_3 \oplus d_4$$

$$p_4 = p_1 \oplus p_2 \oplus p_3 \oplus d_1 \oplus d_2 \oplus d_3 \oplus d_4$$

计算得到

$$p_1 = 0, p_2 = 1, p_3 = 0, p_4 = 0$$

发送方发送:

$$d_1 d_2 d_3 d_4 p_1 p_2 p_3 p_4 = 10010100$$

接收方收到:

$$d_1 d_2 d_3 d_4 p_1 p_2 p_3 p_4 = 10010100$$

接收方再次计算海明码, 得到:

$$p_1 = 0, p_2 = 0, p_3 = 1, p_4 = 1$$

计算两个海明码的异或:

$$s_1 = 0, s_2 = 1, s_3 = 1, s_4 = 1$$

因为  $s_4 = 1$  且  $s_3 s_2 s_1$  不为 0, 所以是 **1 位** 出错。

因为  $s_1 = 0$ , 所以  $d_1, d_2, d_4$  没有出错。所以 **d3 出错**, 将  $d_3$  翻转即可。