

一、选择（满分 10 分，一个题 1 分）

1. 划分计算机软件和硬件的界面是\_\_\_\_\_ B \_\_\_\_\_。  
(A) 操作系统 (Operating System)  
(B) 指令集 (Instruction Set)  
(C) 内存 (Memory)  
(D) 编译器 (Compiler)
2. 对 8 位二进制数，下列说法正确的是\_\_\_\_\_ B \_\_\_\_\_  
(A) -127 的补码为 10000000  
(B) -127 的反码等于 0 的移码  
(C) +1 的移码等于 -127 的反码  
(D) 0 的补码等于 -1 的反码
3. 计算机中采用补码运算的目的是为了\_\_\_\_\_ C \_\_\_\_\_。  
(A) 与手工运算方式保持一致  
(B) 提高运算速度  
(C) 简化计算机的设计  
(D) 提高运算的精度
4. 在定点数运算中产生溢出的原因是\_\_\_\_\_ D \_\_\_\_\_。  
(A) 运算过程中最高位产生了进位或借位  
(B) 参加运算的操作数超出了机器表示的范围  
(C) 寄存器的位数太少，不得不舍弃最低有效位  
(D) 运算的结果超出了机器的表示范围
5. 在浮点数加减法的对阶过程中，\_\_\_\_\_ D \_\_\_\_\_。  
A. 将被加（减）数的阶码向加（减）数的阶码看齐  
B. 将加（减）数的阶码向被加（减）数的阶码看齐  
C. 将较大的阶码向较小的阶码看齐  
D. 将较小的阶码向较大的阶码看齐
6. 在定点二进制运算器中，减法运算一般通过 \_\_\_\_\_ D \_\_\_\_\_ 来实现  
(A) 原码运算的二进制减法器  
(B) 补码运算的二进制减法器  
(C) 补码运算的十进制加法器  
(D) 补码运算的二进制加法器
7. 在计算机中，存放微指令的控制存储器属于\_\_\_\_\_ D \_\_\_\_\_。  
A. 外存  
B. 高速缓存  
C. 内存  
D. CPU
8. 在取指周期中，是按照\_\_\_\_\_ C \_\_\_\_\_的内容访问主存，以读取机器指令。

- A. 指令寄存器 IR
- B. 程序状态字寄存器 PSW
- C. 程序计数器 PC
- D. 主存数据缓冲寄存器 MBR

9. 微程序控制器中，机器指令与微指令的关系是     B    。
- A. 每一条机器指令由一条微指令来执行
  - B. 每一条机器指令由一段微指令编写的微程序来解释执行
  - C. 每一条机器指令组成的程序可由一条微指令来执行
  - D. 一条微指令由若干条机器指令组成
10. 指令系统中采用不同寻址方式的主要目的是     C    。
- A. 简化指令译码
  - B. 提高访存速度
  - C. 缩短指令字长，扩大寻址空间，提高编程灵活性
  - D. 实现程序控制

## 二、填空（满分 20 分，一个题 2 分）

1. 冯·诺依曼计算机体系结构的基本思想是：

\_\_\_\_\_。

（答案：将指令和数据一起存储到计算机中，能自动取出并执行，知道程序执行完毕）

2. 就取得操作数的速度而言，寻址方式中速度最快的是 \_\_\_\_\_，  
速度最慢 \_\_\_\_\_。

（答案：立即数寻址，间接寻址）

3. 提高加法器运算速度的关键是 \_\_\_\_\_，  
（答案是：降低进位信号的传播时间）

4. 某机机器字长 24 位，指令字长 24 位，定长操作码，共能完成 130 种操作，采用单地址格式可直接寻址的范围是 \_\_\_\_\_，变址寻址范围是 \_\_\_\_\_。

（答案是： $2^{16}$ ， $2^{24}$ ）

5. 在指令格式设计中，采用扩展操作码技术的目的是：

\_\_\_\_\_。

（答案：在指令定长的情况下，增加指令数量）

6. 寄存器间接寻址方式中, 操作数存放在\_\_\_\_\_, 寄存器中存放的是\_\_\_\_\_。

(答案: 主存, 操作数地址)

7. 在浮点数表示方法中, 阶码表示\_\_\_\_\_

(答案: 小数点的位置)

阶码位数越多, 该浮点数表示的\_\_\_\_\_ 越大。

(答案: 范围)

8. 采用数据校验码的目的是\_\_\_\_\_。

(答案: 检查并纠正数据在传输中的错误)

9. 常用的数据校验码有奇偶校验码、\_\_\_\_\_和\_\_\_\_\_。

(答案: 海明校验码和循环冗余校验码)

在微程序控制中, 一个节拍中所需要的一组微命令, 被编成一条\_\_\_\_\_微指令\_\_\_\_\_。

10. 移码主要用于表示\_\_\_\_\_。

(答案: 浮点数的阶码部分)

### 三、简答题 (满分 32 分, 每题 8 分)

1. 以四位并行加法器为例, 简述加法运算器中快速进位链的作用及其实现原理

2. 简述运算器的组成及功能。

3. 简述计算机设计步骤, 及各步中完成的主要工作。

4. 如何判断浮点加减运算是否溢出?并说明发生溢出时如何处理?

### 四、综合题 (共 38 分)

1. (10 分) 已知机器字长 8 位,  $x = -0.01111$ ,  $y = +0.11001$ , 求  $[x]_{\text{补}}$ ,  $[-x]_{\text{补}}$ ,  $[y]_{\text{补}}$ ,  $[-y]_{\text{补}}$ ,  $x + y = ?$   $x - y = ?$  要求给出运算器的计算过程, 并用溢出判别方法判断结果

是否溢出。

答案:

$$[x]_{\text{补}} = 1.100\ 0100, [-x]_{\text{补}} = 0.011\ 1100$$

$$[y]_{\text{补}} = 0.110\ 0100, [-y]_{\text{补}} = 1.001\ 1100$$

$$[x + y]_{\text{补}} = [x]_{\text{补}} + [y]_{\text{补}} = 0.010\ 1000$$

$$\text{溢出} = S_x S_y \overline{S_f} + \overline{S_x} \overline{S_y} S_f = 0, \text{无溢出。}$$

$$[x + y]_{\text{原}} = 0.010\ 1000, x + y = 0.0101。$$

$$[x - y]_{\text{补}} = [x]_{\text{补}} + [-y]_{\text{补}} = 0.110\ 0000$$

$$\text{溢出} = S_x S_y \overline{S_f} + \overline{S_x} \overline{S_y} S_f = 1, \text{溢出。}$$

$x - y$  机器溢出, 无法求得。

2. 假设指令字长为 16 位, 操作数的地址码为 6 位, 指令有零地址、一地址、二地址三种格式。(9 分)

(1) 设操作码固定, 若零地址指令有 M 种, 一地址指令有 N 种, 则二地址指令最多有几种?

(2) 采用扩展操作码技术, 二地址指令最多有几种?

(3) 采用扩展操作码技术, 若二地址指令有 P 条, 零地址指令最多有 Q 条, 则一地址指令最多有几种?

答案:

(1) 操作码 4 位, 共  $2^4$  种操作, 则二地址指令最多有  $2^4 - M - N$  种。

(2) 最多有 15 种。

(3)  $(2^4 - P) \times 2^6 - Q / 2^6$

3. 设 R 寄存器  $(R) = 2000$ ,  $(1000) = 2000$ ,  $(2000) = 3000$

若:  $(PC) = 3000$ , 问在以下寻址方式下访问到的操作数的值什么? (9 分)

(1) 直接寻址 (2000)

(2) 存储器间接寻址 ((1000))

(3) 相对寻址 -2000 (PC)

答案:

(1)  $\text{Data} = (2000) = 3000$

(2)  $\text{Data} = ((1000)) = 3000$

(3)  $\text{EA} = (PC) - 2000 = 3000 - 2000 = 1000$      $\text{Data} = (\text{EA}) = (1000) = 2000$

4. 已知  $X=0.1010, Y=-0.1101$ , 用原码一位乘法计算  $X*Y=?$  要求写出计算过程。(10 分)

解:  $[X]_{\text{原}} = \underline{\hspace{2cm}}$                        $[Y]_{\text{原}} = \underline{\hspace{2cm}}$   
 $[X*Y]_{\text{原}} = \underline{\hspace{2cm}}$                        $X*Y = \underline{\hspace{2cm}}$

实现的具体过程:

答案:

解:  $[X]_{\text{原}} = \underline{0.1010}$                        $[Y]_{\text{原}} = \underline{1.1101}$

若  $[X*Y]_{\text{原}} = z_0.z_1\dots\dots z_8$

则  $z_0 = 0\oplus 1 = 1$

实现的具体过程:

C	P	Y	说明(不用写)
0	0 0 0 0	1 1 0 1	开始, 设 $P_0=0$
	<u>+1 0 1 0</u>		$y_4=1, +X$
0	1 0 1 0		C,P 和 Y 同时右移一位
0	0 1 0 1	0 1 1 0	得 $P_1$
			$y_3=0$ , 不作加法
0	0 0 1 0	1 0 1 1	C,P 和 Y 同时右移一位
	<u>+1 0 1 0</u>		$y_2=1, +X$
0	1 1 0 0		C,P 和 Y 同时右移一位
0	0 1 1 0	0 1 0 1	得 $P_3$
	<u>+1 0 1 0</u>		$y_1=1, +X$
1	0 0 0 0		C,P 和 Y 同时右移一位
0	1 0 0 0	0 0 1 0	得 $P_4$

$z_1\dots\dots z_8 = 10000010$

$[X*Y]_{\text{原}} = \underline{1.10000010}$                        $X*Y = \underline{-0.10000010}$