
第一章思考题答案

1. 基于总线结构的计算机系统通常由哪 5 个部分构成？并简述各部分的主要作用。

解答：

1. 中央处理器 CPU(central processor unit)或称微处理器(microprocessor unit)

中央处理器具有算术运算、逻辑运算和控制操作的功能，是计算机的核心。

2. 总线

总线是把计算机各个部分有机地连接起来的导线，是各个部分之间进行信息交换的公共通道。

3. 存储器(memory)

存储器的功能是存储程序、数据和各种信号、命令等信息，并在需要时提供这些信息。

4. 输入输出 (I/O) 接口

外部设备与 CPU 之间通过输入输出接口连接。

5. 输入输出 (I/O) 设备

输入设备是变换输入信息形式的部件。它将人们熟悉的信息形式变换成计算机能接收并识别的信息形式。

输出设备是变换计算机的输出信息形式的部件。它将计算机处理结果的二进制信息转换成人们或其他设备能接收和识别的形式，如字符、文字、图形等。

2. 试举例说明计算机进行加法运算的工作过程。

解答：

示例如下：

```
inta,b,c;
```

```
c=a+b;
```

工作过程简述：a，b，c 都为内存中的数据，CPU 首先需要从内存中分别将 a，b 的值读入寄存器中，然后再执行加法运算指令，加法运算的结果暂存在寄存器中，因此还需要执行数据存储器指令，将运算结果保存到内存中，因此像上例中的 C 语言语句，实际上需要经过两条数据读取指令，一条加法运算指令，一条数据存储器指令才能完成。

3. “冯·诺依曼型结构”计算机与哈佛结构计算机的差别是什么？各有什么优缺点？

解答：

冯·诺依曼结构计算机具有以下几个特点：

- ①有一个存储器；
- ②有一个控制器；
- ③有一个运算器，用于完成算术运算和逻辑运算；
- ④有输入和输出设备，用于进行人机通信；
- ⑤处理器使用同一个存储器存储指令和数据，经由同一个总线传输。

哈佛结构计算机：

①使用两个独立的存储器模块，分别存储指令和数据，每个存储模块都不允许指令和数据并存；

②具有一条独立的地址总线和一条独立的数据总线，利用公用地址总线访问两个存储模块（程序存储模块和数据存储模块），公用数据总线则被用来完成程序存储模块或数据存储

模块与 CPU 之间的数据传输;

③两条总线由程序存储器和数据存储器分时共用。

冯结构优点: 计算机系统结构简单, 只需一套总线; 缺点: 易出现访问内存冲突的问题

哈结构优点: 部分解决了访问内存冲突的问题, 提高了程序执行效率; 缺点: 计算机系统结构复杂。

4. 完成下表中的不同数制之间的转换

二进制 (B)	十进制	十六进制
00001010	10	0x0A
11111111	255	0xFF
1010 0101	165	0xA5
1111 1011 0111 0001	64369	0xFB71
00111111	63	0x3f
0001001101001010	4938	0x134a

5. 实现下表中符号数与机器数补码之间的转换

符号数	机器数补码 (十六进制)
-12	0xF4
+12	0x0C
-256	0xFF00
+256	0x100

6. 分别画出采用小字节序和大字节序的计算机系统存储数据 0x12345678 到地址为 0x00000004 内存空间中的内存映像。

解答:

内存地址	0x00000004	0x00000005	0x00000006	0x00000007
数据	0x78	0x56	0x34	0x12

小字节序

内存地址	0x00000004	0x00000005	0x00000006	0x00000007
数据	0x12	0x34	0x56	0x78

大字节序

7. 写出下列数据的 ASCII 码: 12, 34, 0123, 2012

解答:

12: 0x3132

34: 0x3334

0123: 0x30313233

2012: 0x32303132

8. 采用二进制完成下列无符号数的运算, 并指出是否产生了进位或借位。

0x34+0x98

0x34-0x98

0x1345+0xffff

0x1345-0xffff

解答:

0x34+0x98=0xcc

00110100
+10011000无进位
11001100

$$\begin{array}{r}
 00110100 \\
 0x34-0x98=0x9c-10011000 \text{ 有借位} \\
 10011100
 \end{array}$$

$$\begin{array}{r}
 0001001101000101 \\
 0x1345+0xffff=0x1344 \quad + \quad 1111111111111111 \text{ 有进位} \\
 \hline
 10001001101000100
 \end{array}$$

$$\begin{array}{r}
 0001001101000101 \\
 0x1345-0xffff=0x1346 \quad - \quad 1111111111111111 \text{ 有借位} \\
 \hline
 10001001101000110
 \end{array}$$

9. 采用补码完成下列数的运算，并指出是否产生了溢出。

$$0x34+0x98 \qquad 0x34-0x98 \qquad 0x1345+0xffff \qquad 0x1345-0xffff$$

解答：

由上面的计算过程可知：

$0x34+0x98$ $C_5=0, C_V=0$, 因此没有溢出

$0x34-0x98$ $C_5=1, C_V=0$, 因此有溢出

$0x1345+0xffff$ $C_5=1, C_V=1$, 因此没有溢出

$0x1345-0xffff$ $C_5=1, C_V=1$, 因此没有溢出

10. 请将下列实数用单精度浮点数表示：1.34，345.6，并列式算式 $1.34+345.6$ $345.6-1.34$ 采用单精度数表示的计算过程。

解答：1.34, 345.6 采用二进制表示都无法精确表示，这里采用去尾法略掉满足精度要求之后的各位。

1.34 的二进制数为：1.01010111000010100011110，尾数域的值 01010111000010100011110，指数为 0，加上偏移，因此其二进制值为：01111111，符号位为 0，32 位单精度二进制数表示为：0 0111 1111 01010111000010100011110，采用 16 进制表示为 0x3FAB851E。

345.6 的二进制数为：101011001.100110011001100，规范化的表示为 $1.01011001100110011001100 \times 2^8$ ，因此尾数域为 01011001100110011001100，指数域为 10000111，符号位为 0，32 位单精度二进制数表示为：0 1000 0111 01011001100110011001100，采用 16 进制表示为 0x43ACCCCC。

$1.34+345.6$ 首先需要进行指数对齐，1.34 指数域与 345.6 对齐之后，尾数域变为 00000001010101110000101，与 345.6 的尾数域 01011001100110011001100 相加得到

$$\begin{array}{r}
 00000001010101110000101 \\
 +01011001100110011001100 \\
 \hline
 01011010111100001010001
 \end{array}$$

没有进位，且整数部分不变，因此其 16 进制结果为：0x43AD7851，10 进制数值为 346.93997192

$345.6-1.34$

$$\begin{array}{r}
 01011001100110011001100 \\
 -00000001010101110000101 \\
 \hline
 01011000010000101000111
 \end{array}$$

没有借位，且整数部分不变，因此 16 进制结果为：0x43AC2147，10 进制数值为 344.259979248

由此可知，计算机进行浮点运算时，可能产生一定的误差。

11. 已知一大字节序计算机系统内存空间数据如下:

地址	字节 0	字节 1	字节 2	字节 3	字节 4	字节 5	字节 6	字节 7
0x1234 0000	0xbf	0xf4	0xcc	0xcc	0xcc	0xcc	0xcc	0xcd

若在 C 程序中定义以下数据类型:

```
shortint* sint;  
unsigned short* usint;  
char* ct;  
unsigned char* uct;  
int* cint;  
unsignedint* ucint;  
float * fint;  
double * dint;
```

试说明执行以下语句之后*sint, *usint, *ct, *uct, *cint, *ucint, *fint, *dint 的十进制值各是多少?

```
sint=0x12340000;  
usint=0x12340000;  
cint=0x12340000;  
ucint=0x12340000;  
ct=0x12340000;  
uct=0x12340000;  
fint=0x12340000;  
dint=0x12340000;
```

解答:

*sint 为符号短整形数据, 其十六进制数据为 0xbff4, 相应的十进制值为-16396;
*usint 为无符号短整形数据, 其十六进制数据为 0xbff4, 相应的十进制值为 49140;
*cint 为符号整形数据, 其十六进制数据为 0xbff4cccc, 相应的十进制值为-1074475828;
*ucint 为无符号整形数据, 其十六进制数据为 0xbff4cccc, 相应的十进制值为 3220491468;
*ct 为符号字节形数据, 其十六进制数据为 0xbf, 相应的十进制值为-65;
*uct 为无符号字节形数据, 其十六进制数据为 0xbf, 相应的十进制值为 191;
*fint 为单精度浮点数, 其十六进制数据为 0xbff4cccc, 相应的十进制值为-1.9124999;
*dint 为双精度浮点数, 其十六进制数据为 0xbff4cccccccccd; 相应的十进制值为-1.3

下面提供一种实验验证方法:

基于 PLB 总线 MicroBlaze 计算机系统写入如下程序段。

```
shortint * sint;  
unsignedshortint * usint;  
int * cint;  
unsignedint * ucint;  
char * ct;  
unsignedchar *uct;  
float *fint;  
double *dint;  
void * p;  
p=malloc(8);
```

```

sint=p;
usint=p;
ct=p;
uct=p;
cint=p;
ucint=p;
fint=p;
dint=p;

```

并且在调试过程中，运行完以上指令之后，对 P 所指示的连续 8 个内存单元按题目要求赋值如下：

0x87001db0 : 0x87001DB0 <Hex>		
Address	0 - 3	4 - 7
87001DB0	BFF4CCCC	CCCCCCCC

然后再观察各个不同类型指针指示的数据结果，可得到如下结果。

▲ ➡ sint	0x87001db0
(x)= *sint	-16396
▲ ➡ usint	0x87001db0
(x)= *usint	49140
▲ ➡ cint	0x87001db0
(x)= *cint	-1074475828
▲ ➡ ucint	0x87001db0
(x)= *ucint	3220491468
▲ ➡ ct	0x87001db0
(x)= *ct	-65
▲ ➡ uct	0x87001db0
(x)= *uct	191
▲ ➡ fint	0x87001db0
(x)= *fint	-1.9124999
▲ ➡ dint	0x87001db0
(x)= *dint	-1.3

1.12 计算下列表达式的值：

- (1) unsigned int型数据：1+4294967295=? ； 1-4294967295=?
- (2) int 型数据：2147483647+1=? ； -2147483648-1=?

解答：

- (1) 4294967295=0xffff ffff;4294967295+1=0xffff ffff+0x1=0;
1-4294967295=0x1-0xffff ffff=0x2=2
- (2) 2147483647=0x7fff ffff 为最大的有符号正数，2147483647+1=0x8000 0000，为最小的有符号负数，因此其结果为：-2147483648；
-2147483648=0x8000 0000； -2147483648-1=0x7fff ffff； 因此是 2147483647