

1.2 选择题

(1) 冯 诺依曼结构计算机中数据采用二进制编码表示，其主要原因是？ (D)

1. 二进制运算规则简单
2. 制造两个稳态的物理器件较为容易
3. 便于逻辑门电路实现算数运算

A.1,2 B.1,3 C.2,3 D.1,2,3

分析：冯诺依曼机中的数据采用二进制编码表示，是因为 1.二进制运算规则简单，只有 0 和 1 两种状态，适合采用布尔代数的方法实现运算电路；2.制造两个稳态的物理器件较为容易，更容易用物理状态实现；3.便于逻辑门电路实现算术运算。因为二进制编码只有两种状态，逻辑门电路一般也只有 1 和 0 两种状态。

(2) 下列关于冯 诺伊曼结构计算机基本思想的叙述中，错误的是：(C)

- A. 程序的功能都通过中央处理器执行指令实现
- B. 指令和数据都用二进制表示，形式上并无差别
- C. 指令按地址访问，数据都在指令中直接给出
- D. 执行指令前，指令和数据都需要预先存放在存储器中

分析：程序的功能都会通过计算机内部的控制器和运算器实现，而控制器和运算器又合称为中央处理器（CPU），A 正确；无论是指令还是数据，在计算机中都以二进制形式表示，形式上无差别，B 正确；在指令按照地址访问时，需要通过访问存储器的形式来获得数据，并不是在指令中直接给出的，C 错误；无论是指令还是数据，在指令执行前都会存放在存储器中，D 正确。

(3) 将高级语言源程序转换为机械级目标代码文件的程序成为： (C)

- A. 汇编程序
- B. 链接程序
- C. 编译程序
- D. 解释程序

分析：语言翻译程序主要包括编译程序、汇编程序、解释程序和其他软件操作程序。其中，编译程序负责将高级语言翻译成汇编代码，也称为编译器；汇编程序负责将汇编语言翻译成机器语言目标程序，也称为汇编器；解释程序用于将源程序中的语句按执行顺序逐条翻译成机器指令并执行，且不生成目标程序，也成为解释器。

(4) 计算机硬件能够直接执行的是：(A)

1. 机器语言程序
2. 汇编语言程序
3. 硬件描述语言程序

A.1

B.1,2

C.1,3

D.1,2,3

分析：计算机硬件只能读懂机器语言，因为它们是以二进制形式表示的，直接映射到计算机的硬件结构中。因此其能够直接执行的就是机器语言程序。对于汇编语言程序，还需要经过汇编程序翻译成机器语言程序才能被计算机硬件所执行。硬件描述语言是一种描述数字电路的语言，也需要经过工具转换成低级指令才能被硬件所执行。

(5) 下列选项中，描述浮点数操作速度指标的是 (D)

- A. MIPS
- B. CPI
- C. IPC
- D. MFLOPS

分析：MFLOPS 是指计算机每秒执行浮点运算的次数，而不是 MIPS 所衡量的每秒执行的指令条数。如某系统的运算速度为 2MFLOPS，表示该系统的浮点运算次数每秒 200 万次。更大的单位还有 GFLOPS、TFLOPS、PFLOPS。A：MIPS 即每秒百万条指令，可以以此作为衡量计算机性能的一个指标；B：CPI 指执行每条指令所需要的平均时钟周期数；C：IPC 指每个时钟周期 CPU 能执行的指令条数。

(6) 下列选项中能缩短执行时间的措施是 (D)

- 1. 提高 CPU 时钟频率
- 2. 优化数据结构
- 3. 对程序进行编译优化

- A.1, 2
- B.1,3
- C.2,3
- D.1,2,3

分析：1.提高 CPU 时钟频率可以使得 CPU 在单位时间内执行更多的指令，从而缩短程序执行时间；2.通过优化数据结构，可以减少程序在执行过程中的数据访问时间，从而提高程序的执行效率，进而缩短执行时间；3.编译优化可以使得程序在运行时更加高效，通过减少不必要的指令、减少内存访问次数等方式来提高程序的执行效率。

(7) 某计算机主频为 1.2GHz,其指令分四类，它们在基准程序中所占比例以及 CPI 如图所示，该机的 MIPS 数是？ (C)

表 1.7 各类指令在基准程序中所占比例及 CPI

指令类型	所占比例	CPI	指令类型	所占比例	CPI
A	50%	2	C	10%	4
B	20%	3	D	20%	5

- A.100
- B.200
- C.400
- D.600

分析：

$$\text{根据 } MIPS = \frac{f}{CPI}$$

$$\begin{aligned}\text{有 } MIPS_{总} &= \frac{1.2GHz}{2} \times \frac{1}{2} + \frac{1.2GHz}{3} \times \frac{1}{5} + \frac{1.2GHz}{4} \times \frac{1}{10} + \frac{1.2GHz}{5} \times \frac{1}{5} \\ &= 458\end{aligned}$$

(8) 假定基准程序 A 在某计算机上的运行时间为 100 秒，其中 90 秒为 CPU 时间，其余为 I/O 时间。若 CPU 速度提高 50%，I/O 速度不变，则运行基准程序 A 所耗费的时间是？ (D)

A. 55s

B. 60s

C. 65s

D. 70s

分析：

总指令数为 t ，CPU 计算速度为 v

$$\text{有 } \frac{t}{v} = 90s$$

速度提升 50% 后

$$\frac{t}{1.5v} = \frac{2}{3} \times 90s = 60s$$

$$\text{总时间为 } 60s + 40s = 100s$$

(9) 程序 P 在机器 M 上执行的时间是 20s，编译优化后，P 执行的指令数减少到原来的 70%，而 CPI 增加到原来的 1.2 倍，则 P 在 M 上的执行时间是？ (D)

A. 8.4s

B. 11.7s

C. 14.0s

D. 16.8s

分析：

设程序 P 有 t 条指令

$$\text{有: } t \cdot \text{CPI} = 20s$$

设执行时间为 T

$$T = 1.2 \cdot \text{CPI} \cdot 0.7t = 1.2 \times 0.7 \times 20 = 16.8s$$

(10) 假定计算机 M1 和 M2 具有相同的指令集体系结构，主频分别为 1.5GHz 和 1.2GHz。在 M1 和 M2 上运行某基准程序 P，平均 CPI 分别为 2 和 1，则程序 P 在 M1 和 M2 上的运行时间比值为？ (C)

A. 0.4

B. 0.625

C. 1.6

D. 2.5

分析：

$$f_1 = 1.5 \text{ GHz}, f_2 = 1.2 \text{ GHz}$$

$$\text{则 } MIPS_1 = \frac{f_1}{CPI_1} = \frac{1.5}{2}$$

$$MIPS_2 = \frac{f_2}{CPI_2} = \frac{1.2}{1}$$

∴ 运行时间比值为

$$\frac{t_1}{t_2} = \frac{MIPS_2}{MIPS_1} = \frac{1.2}{\frac{1.5}{2}} = 1.6.$$

1.5 假定某计算机 1 和计算机 2 以不同的方式实现了相同的指令集。该指令集中共有 A,B,C,D 类指令，他们占比分别是 40%,20%,15%,25%，计算机 1 和计算机 2 的时钟频率分别为 600MHz 和 800MHz，各类指令在两计算机上的 CPI 如图所示

表 1.8 各类指令在两计算机上的 CPI

指令类型	A	B	C	D
CPI1	2	3	4	5
CPI2	2	2	3	4

求两计算机的 MIPS 各是多少？

$$CPI_1 = 2 \times 0.4 + 3 \times 0.2 + 4 \times 0.15 + 5 \times 0.25 = 3.25$$

$$\text{则 } MIPS_1 = \frac{f_1}{CPI_1} = \frac{600}{3.25} = 185$$

$$CPI_2 = 2 \times 0.4 + 2 \times 0.2 + 3 \times 0.15 + 4 \times 0.25 = 2.65$$

$$\text{则 } MIPS_2 = \frac{f_2}{CPI_2} = \frac{800}{2.65} = 302$$

1.6 若某程序编译后生成的目标代码由 A,B,C,D 四类指令组成，他们在计算机中所占比例分别为 40%,20%,15%,25%。已知 A,B,C,D 四类指令的 CPI 分别为 1,2,2,2。现需要对程序进行编译优化，优化后的程序中 A 类指令数量减少了一半，而其他指令未发生变化。假设运行该程序的计算机 CPU 主频为 500MHz

(1) 优化前后程序的 CPI 各为多少

$$\text{1) 优化前: } CPI = 1 \times 0.4 + 2 \times 0.2 + 2 \times 0.15 + 2 \times 0.25 = 1.6$$

优化后: 没优化前 A 指令有 0.4t 条

则指令总条数为 t 条

B: 0.2t

C: 0.15t

D: 0.25t

此时 A 减少了一半，剩下 0.2t 条，总条数为 $t - 0.2t = 0.8t$

$$\text{∴ A,B,C,D 指令占比为: } \frac{0.2t}{0.8t} = \frac{1}{4}, \frac{0.2t}{0.8t} = \frac{1}{4}, \frac{0.15t}{0.8t} = \frac{3}{16}, \frac{0.25t}{0.8t} = \frac{5}{16}$$

$$\text{∴ 优化后的 } CPI = 1 \times \frac{1}{4} + 2 \times \frac{1}{4} + 2 \times \frac{3}{16} + 2 \times \frac{5}{16} = 1.75$$

(2) 优化前后程序的 MIPS 各为多少

$$\begin{aligned} \text{(2) 根据 } \text{MIPS} &= \frac{f}{\text{CPI}} \\ \text{优化前: } \text{MIPS} &= \frac{500 \times 10^6}{1.6 \times 10^6} = 312.5 \\ \text{优化后: } \text{MIPS} &= \frac{500 \times 10^6}{1.76 \times 10^6} = 284.7 \end{aligned}$$

(3) 通过上面的计算你能得出什么结论

答: 优化后 CPI 增加, MIPS 减少, 并没有真正地实现优化, 应该尝试降低 CPI 最大指令的比例。

2.2

(1) 由三个“1”和五个“0”组成的八位二进制补码, 能表示的最小整数是 (B)

- A. -126
- B. -125
- C. -32
- D. -3

分析: 补码最高为表示符号位, 我们要求出最小整数, 需要计算出补码转换后的真值的最小值。首先 1 放在最高位表示负数, 还剩两个 1 和五个 0, 计算其真值需要取反+1, 因此取反后为两个 0 和五个 1, 将五个 1 放在高位, 再+1 即可表示最小值。真值为 111 1101, 转换成十进制为 125, 最后加上符号位, 即为-125。

1000 0011 -> 1111 1101 -> -125

(2) 考虑以下 c 语言代码:

```
Unsigned short usi=65535;
```

```
Short si=usi;
```

执行上述程序后, si 的值是 (A)

- A. -1
- B. -32767
- C. -32768
- D. -65535

分析: 65535 转换为二进制为 1111 1111 1111 1111, 对于无符号数, 其在计算机内部的存储形式是二进制。将这个数值赋给有符号数后, 最高位表示符号位, 表示负数。然后再计算数值位: 将最高位后面所有位按位取反再+1 得到了 000 0000 0000 0001, 结合符号位, 最终结果为-1。

(3) 假定编译器规定 int 和 short 类型长度为 32 位和 16 位, 执行下列 C 语言语句 unsigned short x=65530; unsigned int y=x; 得到的 y 的机器数为 (B)

- A. 0000 7FFAH
- B. 0000 FFFAH
- C. FFFF 7FFAH
- D. FFFF FFFAH

分析：将 65530 转换为十六进制为：FFFAH。再将其从 short 类型转换成 int 类型，属于小字长转大字长，而 y 又是无符号 int 类型，在该转换过程中会进行零扩展，高位自动填充 0，因此 y 的机器数为 0000 FFFAH。

(4) 有如下 C 语言程序段:short si=-32767;unsigned short usi = si;执行上述两条语句后 usi 的值为 (D)

- A. -32767
- B. 32767
- C. 32768
- D. 32769

分析：首先 si 是有符号 short 类型变量，在计算机系统中以补码形式存储，补码形式为：1000 0000 0000 0001。将其赋值给无符号 short 类型变量 usi，根据无符号数在计算机中存储方式为二进制，将 1000 0000 0000 0001 转换成十进制就为 32769。

(5) float 型数据通常使用 IEEE754 单精度浮点数格式表示。若编译器将 float 型变量 x 分欸在一个 32 位浮点寄存器 FR1 中，且 x=-8.25，则 FR1 的内容是 (A)

- A. C104 0000H
- B. C242 0000H
- C. C184 0000H
- D. C1C2 0000H

分析：首先将 x 转换成二进制形式：-1000.01，再将其标准化，标准化后是-1.00001 * 2³。因为是负数，所以符号位为 1，指数位偏移后的值 3 加上偏移量 127 后为 130，尾数即为标准化后的小数部分。因此符号位：1；指数位：130，转成二进制：10000010；尾数位：000010000000000000000000。将三部分组合起来得到 1 10000010 000010000000000000000000，最后转成十六进制得到 C104 0000H。

(6) 某数采用 IEEE754 单精度浮点数格式表示位 C640 0000H,该数的值位 (A)

- A. -1.5x2¹³
- B. -1.5x2¹²
- C. -0.5x2¹³
- D. -0.5x2¹²

分析：对于给定的十六进制数 C640 0000H，先将其转换成二进制：1100 0110 0100 0000 0000 0000 0000 0000。可知符号位是 1，表示负数；指数位是 1000 1100，转换成十进制为 140，再减去偏移量 127 得到 13；剩余均为尾数位，表示小数部分。最后小数部分补上 1，得到 1.1，转换成十进制为 1.5。所以最后的值为 -1.5 * 2¹³。

(7) float 型(IEEE754)能表示的最大正整数是: (D)

- A. 2¹²⁶-2¹⁰³
- B. 2¹²⁷-2¹⁰⁴
- C. 2¹²⁷-2¹⁰³
- D. 2¹²⁸-2¹⁰⁴

分析：要使表示的数最大，符号位一定是 0 表示整数，阶码也得最大。但要注意的是八位阶码不能全为 1，因此最大的阶码为 1111 1110，转成十进制是 254，再减去偏移量 127 得到 127。对于尾数，我们都取 1 即可。因此最大正整数为 2⁰ + 2⁻¹ + 2⁻² + ... + 2⁻²³ = 2 -

2^{23} ，而阶码最大为 127，所以最大正整数为 $2^{127} * (2 - 2^{23}) = 2^{128} - 2^{104}$ 。

(8) IEEE754 单精度浮点格式表示的数中，最小规格化正数是: (A)

A. 1.0×2^{-126}

B. 1.0×2^{-127}

C. 1.0×2^{-128}

D. 1.0×2^{-149}

分析：最小规格化正数的指数部分为-126，位数部分为 1，而规格化的浮点数总是以 1 开始，因此最小规格化正数的表示形式为： $1.0 * 2^{-126}$ 。

(9) float 型数据通常用 IEEE754 单精度浮点格式表示，假定两个 float 型变量 x 和 y 分别存放在 32 位寄存器 f1 和 f2 中，若(f1)=CC90 0000H,(f2)=B0C0 0000H,则 x 和 y 之间的关系为 (A)

A. $x < y$ 且符号相同

B. $x < y$ 且符号不相同

C. $x > y$ 且符号不相同

D. $x > y$ 且符号相同

分析：将 f1 转成二进制：1100 1100 1001 0000 0000 0000 0000 0000，将 f2 转为二进制：1011 0000 1100 0000 0000 0000 0000 0000，可知最高位均为 1，符号相同。再看 f1 指数位：1001 1001，f2 指数位：0110 0001，显然 f1 指数位大于 f2 指数位。而两者又都是负数，所以 $x < y$ 。因此选 A。

(10) 假定变量 i,f,d 的数据类型分别为 int,float,double(补码，IEEE754 单精度双精度) $i=785, f=1.5678e3, d=1.5e100$,若在 32 位计算机中执行下列关系表达式，则结果为真的是 (D)

I $i == (int)(float)f$

II $f == (float)(int)f$

III $f == (float)(double)f$

IV $(d+f)-d == f$

A. 仅 I、II B. 仅 I、III C. 仅 II、III D. 仅 III、IV

分析：对于 A：f 是浮点数，先转成整形，得到 1567，而 i 的值为 785，因此结果为假；对于 B：f 是浮点数，先经过整形转换得到 1567，再转成浮点数为 1567.0，而 f 的值为 1567.8，因此结果为假；对于 C：单精度转双精度再转单精度值不变，因此表达式成立；对于 D：d + f 先得到一个双精度浮点数，再减去 d，能得到 f。

(11) 用海明码对长度为 8 位的数据进行检错和纠错时，若能纠正一位错，则校验位数至少为 (C)

A. 2

B. 3

C. 4

D. 5

分析：校验位数可以通过 $r \geq \log_2(n + r + 1)$ 向上取整得到，其中 n 是数据位的数量，r 是校验位的数量。将 $n = 8$ 带入可得 r 至少为 4。

2.6C 语言中允许无符号数和有符号整数之间的转换，下面是一段 c 语言代码

```
Int x=1;
Unsigned u=2147483648;
Printf("x=%u=%d\n",x,x);
Printf("u=%u=%d\n",u,u);
```

给出在 32 位计算机中上述程序段的输出结果并分析原因.

输出结果如下:

```
x=4294967295=-1;
u=2147483648=-2147483648
```

原因: 1.在计算机中整数以补码形式表示和存储; 2. $x=-1$, 先求-1 的 32 位补码, 机器码是 $2^{32}-1=4294967295$ 。所以第一行输出是机器码和真值; 3. $u = 2^{31}$ 是一个无符号数, 无溢出, 由于首位为 1, %u 输出机器码就是 2147483648, %d 输出是真值, 将该机器码按补码转换成真值, 所以是-2147483648。

2.7 (3)16 位补码定点小数能表示的数据范围?

16 位补码定点小数: 1.000 0000 0000 0000 - 0.111 1111 1111 1111, 转成十进制就为 -1 - 1 - 2^{-15}

2.9 用 IEEE754 32 位单精度浮点数标准表示下列十进制数

(1) -6.625

由于该数为负数, 因此符号位为 1, 将 6.625 转成二进制得到 110.101, 将其标准化后得到总的二进制为 1100 0000 1101 0100 0000 0000 0000 0000, 转成十六进制为 C0D4 0000H

(2)3.1415927

由于该数为正数, 因此符号位为 0, 同上题操作可以得到最终标准化的二进制为 0100 0000 0100 1001 0000 1111 1101 1011, 转成十六进制得到 4049 0FDBH

2.16 由 6 个字符的 7 位 ASCII 字符排列, 再加上水平和垂直偶校验码构成表 2.27 所示行列结构

表 2.27 ASCII 交叉校验

字符	7 位 ASCII 字符							HP
3	0	X_1	X_2	0	0	1	1	0
Y_1	1	0	0	1	0	0	X_3	1
+	X_4	1	0	1	0	1	1	0
Y_2	0	1	X_5	X_6	1	1	1	1
D	1	0	0	X_7	1	0	X_8	0
=	0	X_9	1	1	1	X_{10}	1	1
VP	0	0	1	1	1	X_{11}	1	X_{12}

则 X_1, X_2, X_3, X_4 处的比特分别为 1110
 X_5, X_6, X_7, X_8 处的比特分别为 1000
 $X_9, X_{10}, X_{11}, X_{12}$ 处的比特分别为 1011
 Y_1 和 Y_2 处的字符分别为 1 和 7

分析：先判断 x_4 ，由于水平校验位和竖直校验位均为 0，因此 $x_4 = 0$ ，这也能判断出来水平和竖直方向上均采用偶校验，根据偶校验定义即可快速判断其他各位上的比特值。

2.17 设 8 位有效信息为 01101110，试写出它的海明码，给出过程，说明分组检测方式，并给出指错字及其逻辑表达式，如果接收方收到的有效信息变为 01101111，说明如何定位错误并纠正错误

二进制	1	10	11	100	101	110	111	1000	1001	1010	1011	1100
位置	1	2	3	4	5	6	7	8	9	10	11	12
	P_1	P_2	D_1	P_3	D_2	D_3	D_4	P_4	D_5	D_6	D_7	D_8

分组为 $P_1: D_1, D_2, D_4, D_6, D_7$

$$\therefore P_1 = D_1 \oplus D_2 \oplus D_4 \oplus D_6 \oplus D_7 = 1$$

$P_2: D_1, D_3, D_4, D_6, D_7$

$$\text{同理, } P_2 = 1$$

$P_3: D_2, D_3, D_4, D_8$

$$P_3 = 0$$

$P_4: D_5, D_6, D_7, D_8$

$$P_4 = 1$$

$$\therefore G_1 = P_1 \oplus D_1 \oplus D_2 \oplus D_4 \oplus D_6 \oplus D_7 = 0$$

$$G_2 = P_2 \oplus D_1 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 = 0$$

$$G_3 = P_3 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_8 = 1$$

$$G_4 = P_4 \oplus D_5 \oplus D_6 \oplus D_7 \oplus D_8 = 1$$

$$\therefore \text{指错码: } G_4 G_3 G_2 G_1 = 1100 = 12$$

若一位错，则为 12 位错，取反即可

2.18 设要采用 CRC 码传送数据信息 $x=1001$ ，当生成多项式为 $G(x)=1101$ 时，请写出它的循环冗余码。若接收方收到的信息为 $x'=1101$ ，说明如何定位错误并纠正错误

信息 $x=1001$, 可知 $n=4$

根据 $n+r \leq 2^r-1$ 得到 $r=3$

先将 x 左移 r 位, 得到 1001000

再选择 $r+1$ 位的生成多项式 1101

作模2除法

$$\begin{array}{r} 1111 \\ 1101 \overline{) 1001000} \\ \underline{1101} \\ 000 \\ \underline{1101} \\ 1010 \\ \underline{1101} \\ 1110 \\ \underline{1101} \\ 011 \end{array}$$

余数为 011 , 所以循环码为 1001011

假设接收的信息为 1101011
出错

则对其作模2除法

$$\begin{array}{r} 1000 \\ 1101 \overline{) 1101011} \\ \underline{1101} \\ 0000 \\ \underline{0000} \\ 0001 \\ \underline{0000} \\ 0011 \\ \underline{0000} \\ 011 \end{array}$$

余数为 011 , 将 011 继续补0作除法运算
经过两次运算后余数为 001 , 所以是第2位错
将左侧第2位取反即可.