

1、(9%) 计算:要求给出关键的计算过程。

(a) 将十进制数 -131072 转化为 32 位二进制补码整数, 给出其十六进制表示;

(2%)

(b) 采用二进制补码整数数据类型, 计算 $1101 + 01110111$, 结果以十进制表示, 并判断是否溢出?(2%)

(c) 做二进制逻辑运算: $(\text{NOT}(A1) \text{ AND } x2B) \text{ OR } (xC3 \text{ XOR } xD4)$, 结果以十六进制表示;(2%)

(d) -12.345 的 32 位二进制编码? (IEEE 754 标准)(3%)

3、(13%)

(a) 一个有 9 条输入线的译码器，有多少输出线?(2%)

(b) 一个有 4 条输入线的多路选择器，有多少输出线?有多少选择线? (3%)

(c) 某计算机为按字节寻址(即寻址能力)，访问一个内存单元需要 32 位，(3%)

(i) 该内存的地址空间?

(ii) 该内存的大小为多少位?

(ii) 该内存的大小为多少字节?

(d) 简要描述:冯•诺依曼提出的存储程序控制原理的思想。(5%)

4、(6%)

假设一个机器有如下格式的 48 位指令:

操作码	目标寄存器	源寄存器	立即数
-----	-------	------	-----

如果共有 216 个操作码, 64 个寄存器, 那么, 当立即数表示补码整数类型时, 立即数的最大和最小值分别是多少? 要求给出关键的计算过程。

5、(13%)

```

01 ;
02 ;
03 ;
04 ;
05     .data    x3000000A
06     .align   2
07 numbers: .word  #10, #3, #4, #6, #8, #-2, #45, #5, #8, #9
08 sum:     .space  4
09 ;
0A ;
0B     .text    x40000000
0C     .global  main
0D main:      addi   r1, r0, numbers
0E          addi   r3, r0, #0          ;
0F          addi   r2, r0, #10         ;
10 ;
11 ;
12 again:    beqz   r2, exit
13          lw     r4, 0(r1)
14          add    r3, r3, r4
15          addi   r1, r1, #4          ;
16          subi   r2, r2, #1
17          j      again
18 exit:     sw     sum(r0), r3
19          trap   #0

```

(a) (5%)填写符号表:

标记	地址

(b)(4%)该程序实现了什么功能?程序执行结束时，各个寄存器的值是多少?

寄存器	值
R0	
R1	
R2	
R3	
R4	

(c)(4%)参考下表，将 0D、12 和 17 行的指令翻译为机器语言指令。

指令	操作码	指令	操作码
LHI	001100	SLTI	010000
LW	011100	ADDI	000001
BEQZ	101000	JAL	101110
J	101100	SW	011101

0D:

31	26	25	21	20	16	15	11	10	6	5	0

12:

--	--	--	--

17:

--	--

6、(7%)下面的程序将 R1 中的一个两位数(10~99 之间的整数)显示到屏幕上。

```
01          .DATA
02  OUTBUF :  .SPACE    #40
03  ;
04          .TEXT
05          .GLOBAL  MAIN
06  MAIN :   JAL      A
07          JAL      B
08          TRAP     x00
09  ;
0A  A :      ADDI     R2, R0, #0
0B  LOOP :   SUBI     R1, R1, #10
0C          ADDI     R2, R2, #1
0D          SLTI     R3, R1, #10
0E          BNEZ     R3, EXIT
0F          J        LOOP
10  EXIT :   ADDI     R4, R0, OUTBUF
11          ADDI     R2, R2, x30
12          SB       0(R4), R2
13          ADDI     R1, R1, x30
14          SB       #1(R4), R1
15          SB       #2(R4), R0
16          JR       R31
17  ;
18  B :      TRAP     x08          ; 输出字符串
19          ADDI     R4, R0, #10   ; 输出新行
1A          TRAP     x07
1B          JR       R31
```

(a) 程序不能正常工作，为什么?(3%)

(b) 如何修正该错误?(4%)

7、(12%) 如下代码片段的输出各是什么?

(a) (3%)

```
int i=3;
int j=0;
for (; i >= 0; i--) {
    for (; j <= i; j++)
        printf("+");
    for (; j >= i; j++)
        printf("-");
    printf("\n");
}
```

输出:

(b)(3%)

```
int i;
for ( i = 1; i <= 5; i++)
    switch (i % 5){
        case 0:
            printf("+"); break;
        case 1:
            printf("-"); break;
        default: printf("/");
        case 2: printf("*");
    }
```

输出:

(c) (3%)

```
char str[]="Hello World!\0";
char *p;
p = str;
while ( *p != 0 ) {
    printf( "%c", *p++);
    p++; }

```

输出:

(d)(3%)

```
char str[]="Hello World!\0";
char *p;
p = str;
while ( *p != ' ' ) {
    printf( "%c", *p - 1 );
    p++; }

```

输出:

8、(8%) 下面的程序的功能是将输入的字符串中的所有空白

TAB、空格)删除，并输出结果。请填空((A)~(H))，将程序补充完整。

```
#include <stdio.h>
```

```
void delspace(char *p1, char * p2);
```

```
int main(){
```

```
    char c;
```

```
    char s1[80];
```

```
    char s2[80];
```

```
    int i = 0;
```

```
    c = getchar ();
```

```
    while (c != '\n'){
```

```
        s1[i] = c;
```

```
        _____(A)
```

```
        _____(B)
```

```
    }
```

```
    _____(C)
```

```
    _____(D)
```

```
    printf ("%s\n", s2);
```

```
}
```

```
void delspace(char *p1, char * p2){
```

```
    int i, k;
```

```
    i = 0;
```

```
    k = 0;
```

```
    while (p1[i] != '\0'){
```

```
        if (_____(E)){
```

```
            _____(F)
```

```
            _____(G)
```

```
        }
```

```
        i++;
```

```
    }
```

```
    _____(H)
```

```
}
```

9、(12%)

(a)该 C 函数的功能是什么？

(b)将如下 C 函数翻译为 DLX 汇编代码。(DLX 将 R4~R7 用于参数传递，将 R2 和 R3 用于返回值，将 R16~R23 用于局部变量，将 R8~R15 和 R24、R25 用于存储临时产生的值，R29 为运行时栈指针。)

```
int MyFunc (char * p1){  
    int i;  
    char *p2 = p1;  
    while (*p2)  
        p2++;  
    i = p2 - p1;  
    return i;  
}
```

10、(12%) 编写一个 C 函数: `void Func(char* str)`, 将 `str` 所指

的字符串反转。并使用这个函数写一个程序:从键盘输入一个字符串, 调用 `Func` 函数后输出结果。例如, 如果输入为 “abcd”, 则输出为 “dcba”。