

数制转换，原码、补码、移码转换

1、将 $(15B.A)_{16}$ 转换成二进制、八进制、十进制

$$(15B.A)_{16} = (0001\ 0101\ 1011.1010)_2 = (0533.50)_8 = (347.625)_{10}$$

2、分别求 $(-1110\ 1001\ 101)_2$ 的16位原码、补码和移码及表示范围

$$\begin{aligned} (-1110\ 1001\ 101)_2 &= (1000\ 0111\ 0100\ 1101)_{\text{原}} = (1111\ 1000\ 1011\ 0011)_{\text{补}} \\ &= (0111\ 1000\ 1011\ 0011)_{\text{移}} \end{aligned}$$

原码：- $(2^{15}-1) \sim (2^{15}-1)$

补码：- $(2^{15}) \sim (2^{15}-1)$

移码：- $(2^{15}) \sim (2^{15}-1)$

浮点数与真值之间相互转换

3、求 $(-1101.011)_2 \times 2^{-15}$ 的IEEE754单精度浮点数表示

1 01110011 101 0110 0000 0000 0000 0000

4、求IEEE754单精度浮点数表示0010 1101 0101 1000 0000 0000 0000 0000的真值

$(1.1011)_2 \times 2^{-37}$

5、说明下面func函数的功能。

```
int func (unsigned word)
{
    return (word & (0x07<<3)) >> 3;
}
```

提取一个32位数的3-5位

C程序分析

6、有下列变量定义，请在32位机器上输出si, usi, i, ui的十进制（真值）和十六进制值（机器数）

```
short si = -32768; //32768=215
```

```
unsigned short usi = si;
```

```
int i = si;
```

```
unsigned ui = usi ;
```

真值	机器数
si = -32768	8000H
usi = 32768	8000H
i = -32768	FFFF8000H
ui = 32768	00008000H

7、写出下面程序在32位机器上运行时的结果

```
int x=-1;
```

```
unsigned u=2147483648; // 2147483648=231
```

```
printf( "x=%u=%d\n" ,x,x);
```

```
printf( "u=%u=%d\n" ,u,u);
```

x=4294967295=-1

u=2147483648=-2147483648

补码加减法及寻址方式

8、用补码加减运算方法计算8位无符号数65和100的和，并判断进\借位标志位。

1010 0101 CF=0

9、用补码加减运算方法计算8位带符号数-52和-82的差，并判断溢出标志位。

0001 1110 OF=0

10、某计算机内存采用按字节编址方式。指令固定32位长。假设当前指令的地址为D214H，指令中给出的形式地址为0268H，基址寄存器的内容为248AH，变址寄存器的内容为8。说明以下各种情况下操作数的有效地址分别是多少？

操作数采用基址寻址；

基址：26F2H

操作数采用变址寻址；

变址：0270H

操作数采用相对寻址。

相对：D480H

汇编、机器、C语言相互转换

11、将MIPS汇编表示的指令sub \$8, \$9, \$10翻译成MIPS机器码。

000000 01001 01010 01000 00000 100010

12、某条MIPS指令的二进制代码表示为0000 0000 1010 1111 1000 0000 0010 0000，该指令对应的MIPS汇编表示是什么？

add \$s0,\$a1,\$t7 或 add \$16,\$5,\$15

13、将下列翻译成汇编语言

if (i == j)

f = g+h ;

else

f = g-h ;

假定 i, j, f, g, h分配给\$1, \$2, \$3, \$4, \$5

bne \$1, \$2, else

add \$3, \$4, \$5

j exit

else:

sub \$3, \$4, \$5

exit:

地址计算及数据存放

14、MIPS无条件转移指令“j target address”的地址为0x00400000，指令中存储的26位立即数为0x110BA0。计算该指令的转移目标地址。

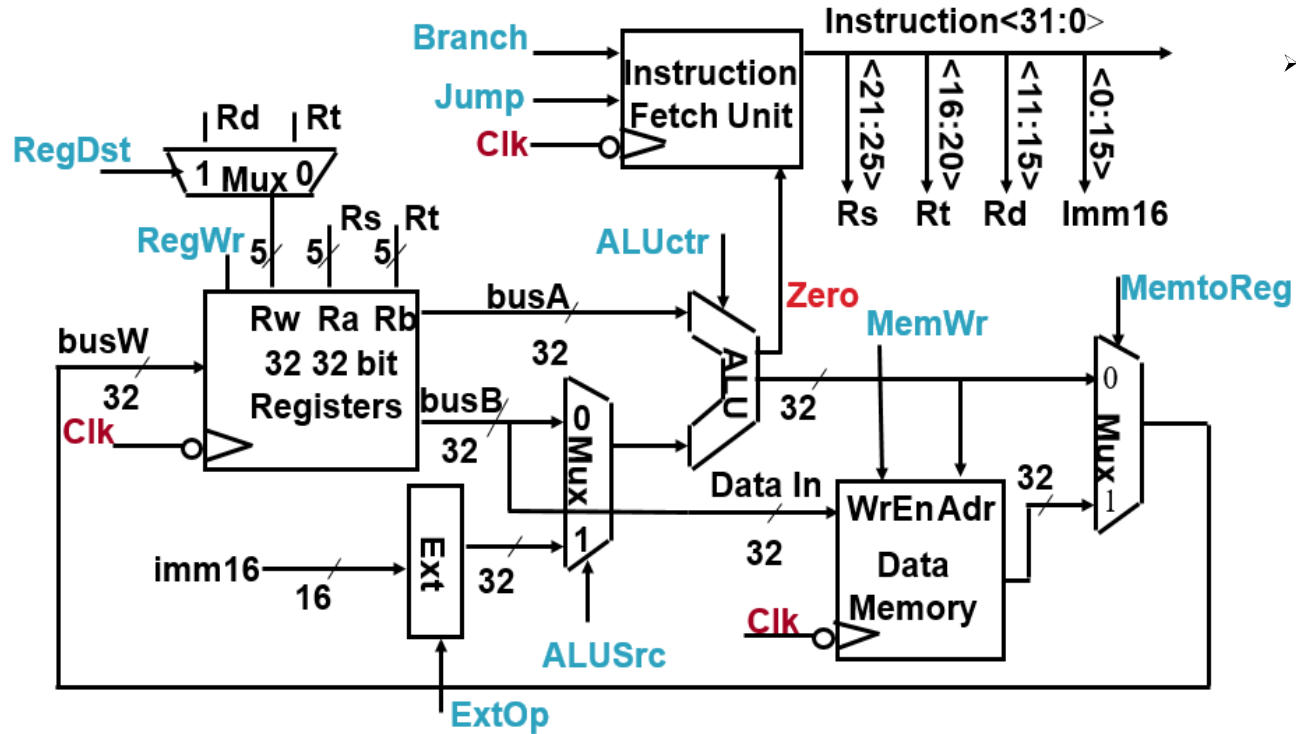
0000 00 0001 0001 0000 1011 1010 0000 00

15、数据0x88在内存中占2个字节，从内存地址0x1000开始存放该数据，若按小端模式进行存放，则地址0x1000地址存放的是_____，地址0x1001存放的是_____。（两个空格都用16进制表示填写）

88H 00H

指令执行过程及控制字

16、分析指令执行过程 (add rd, rs, rt; ori rt, rs, imm16; lw rt, rs, imm16) , 并给出控制信号的取值 (RegDst, RegWr, ALUctr, ExtOp, ALUSrc, MemWr, MemtoReg)



➤ `add rd, rs, rt`

- ✓ `M[PC]` 从PC所指的内存单元中取指令 (公共操作, 取指部件完成)
- ✓ $R[rd] \leftarrow R[rs] + R[rt]$ 从rs、rt 所指的寄存器中取数后相加。若结果不溢出, 则将结果送rd 所指的寄存器中; 若结果溢出, 则不送结果, 并转到“溢出处理程序”执行。
- ✓ $PC \leftarrow PC + 4$ PC加4, 使PC指向下一条指令 (公共操作, 取指部件完成)

**RegDst=1, RegWr=1, ALUctr=add, ExtOp=x,
ALUSrc=0, MemWr=0, MemtoReg=0**

指令执行过程及控制字

◆ori rt, rs, imm16

- $M[PC]$ 取指令 (公共操作, 取指部件完成)
- $R[rt] \leftarrow R[rs] \text{ or } \text{ZeroExt}(\text{imm16})$ 立即数零扩展, 并与rs内容做“或”运算
- $PC \leftarrow PC + 4$ 计算下地址 (公共操作, 取指部件完成)

$\text{RegDst}=0, \text{RegWr}=1, \text{ALUctr}=\text{or}, \text{ExtOp}=0, \text{ALUSrc}=1, \text{MemWr}=0, \text{MemtoReg}=0$

◆lw rt, rs, imm16

- ✓ $M[PC]$ 取指令 (公共操作, 取指部件完成)
- ✓ $\text{Addr} \leftarrow R[rs] + \text{SignExt}(\text{imm16})$ 计算存储单元地址 (符号扩展!)
- ✓ $R[rt] \leftarrow M[\text{Addr}]$ 装入数据到寄存器rt中
- ✓ $PC \leftarrow PC + 4$ 计算下地址 (公共操作, 取指部件完成)

$\text{RegDst}=0, \text{RegWr}=1, \text{ALUctr}=\text{addu}, \text{ExtOp}=1, \text{ALUSrc}=1, \text{MemWr}=0, \text{MemtoReg}=1$

映射关系及中断过程

17、主存空间大小为32GB，按字节编址。Cache的数据区（不包括标记、有效位等存储区）有256KB，块大小为128B，采用二路组直接映射方式，将主存地址划分成标记tag、组索引和块内地址三部分，其中：标记tag为_____位，组索引为_____位，块内地址_____位。

17,11,7

18、阐述中断的过程。

当外设准备好时，便向CPU发中断请求，CPU响应后，中止现行程序的执行，转入一个“中断服务程序”进行输入/出操作，实现主机和外设接口之间的数据传送，并启动外设工作。“中断服务程序”执行完后，返回原被中止的程序断点处继续执行。此时，外设和CPU并行工作。

19、阐述中断的三个条件以及任务。

三个条件：CPU处于“开中断”状态
至少要有有一个未被屏蔽的中断请求
当前指令刚执行完

三个任务：保存被中断的程序断点处的关键性信息
识别中断源并根据中断响应优先级进行判优
调出中断服务程序