**计算机系统（3）期中考试（第1-3章）**

**姓名： 学号： 得分：**

1. 一个程序在一台计算机上运行时需要100秒，其中80秒的时间用于乘法操作，通过将乘法操作的速度改进到只需16秒，从而把程序的运行速度提高到5倍。这里改进性能所使用到的是哪个伟大设计思想 C 。

A.通过预测提高性能 B.通过流水线提高性能

C.加速大概率事件 D.通过并行提高性能

1. 对某一芯片只提高工作电压，则其功耗 A ：

A.提高； B.下降； C.不确定； D.保持不变；

1. 下面的图表代表的是哪条MIPS指令? **（ D ）**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| op | Rs | rt | rd | shamt | funct |
| 0 | 8 | 9 | 10 | 0 | 34 |

A. sub $t0, $t1, $t2 B. add $t2, $t0, $t1

C. sub $t2, $t1, $t0 D. sub $t2, $t0, $t1

1. 假设$s1中的值是0xD0000000,给定下列$s0的值,执行下列指令是否会产生溢出？

判断依据：

1. 两个正数相加得到负数，则溢出
2. 两个负数相加得到正数，则溢出
3. 一正一负相加不会溢出
4. 减法转变成加法来判断溢出

**（ B ）**（1）$s0=0x70000000, 执行add $s0,$s0,$s1

A.会溢出 B.不会溢出

**（ B ）**（2）$s0=0x80000000, 执行sub $s0,$s0,$s1

A.会溢出 B.不会溢出

**（ A ）**（3）$s0=0x7FFFFFFF, 执行sub $s0,$s0,$s1

A.会溢出 B.不会溢出

1. 假设$t0=0xBEADFEED, $t1= 0xDEADFADE. 求执行下面指令后寄存器$t2的值。

andi会零扩展(中文版附录中没有，但英文版中有)

1. sll $t2, $t0, 4

or $t2, $t2, $t1 则$t2= **（ 0xFEFF FEDE ）**

（2） sll $t2, $t0, 4

andi $t2, $t2, -1 则$t2= **（ 0xEED0 ）**

（3） srl $t2, $t0, 3

andi $t2, $t2, 0xFFEF 则$t2= **（ 0xBFCD ）**

1. 如果要将乘法指令结果的高32位保存在$t1，低32位保存在$t2中，需要使用两条指令来完成，它们是  **mfhi $t1; mflo $t2**
2. 为了调用函数myfunc1，应该使用指令 **jal myfunc1 ，**函数返回时应该使用指令 **jr $ra** 返回到调用函数处的下一跳指令。
3. 如果指令“beq $t0,$t1, 32”指令位于0x1000地址，执行该指令时PC= **0x1004 ，**若$t0=16,$t1=16，则下一条被执行的指令位于 **0x1084** ：
4. 把下面的C代码翻译成MIPS代码。假设变量f,g,h,i和j分别赋值给寄存器$s0,$s1,$s2,$s3和$s4。假设数组A和B的基地址分别存放在$s6和$s7中。假设数组A和B中的元素均为4字节的字：

1）B[8]=A[i]+A[j];

2）f=g-A[B[4]];

1） 答：

sll $t1, $s3, 2

sll $t2, $s4, 2

add $t1, $t1, $s6

add $t2, $t2, $s6

lw $t1,0($t1)

lw $t2,0($t2)

add $t1, $t1, $t2

sw $t1, 32($s7)

2) 答：

lw $t1,16($s7)

sll $t1, $t1, 2

add $t1, $t1, $s6

lw $t1, 0($t1)

sub $s0, $s1, $t1

1. 将以下C语句转换为MIPS汇编指令序列，假设变量a, b, i, j分别对应寄存器$s0, $s1, $t0, $t1, $s2保存着数组D的起始地址。

a）for ( i=0; i<10; i++)

a += b;

b) while (a<10) {

D[a] = b + a;

a += 1;

}

1. 答：

addi $t0, $zero, 0

Loop: slti $t2, $t0, 10 // $t2 = 1 if $t0 < 10 else 0

beq $t2, $zero, Exit // go to Exit if $t2 == 0

add $s0, $s0, $s1 // a += b

addi $t0,$t0,1 // i++

j Loop

Exit:

1. 答：

Loop:

slti $t2, $s0, 10 // $t2 = 1 if $s0(a) < 10

beq $t2, $zero, Exit // go to Exit if $t2 = 0

add $t2, $s0, $s1 // $t2 = b + a

sll $t3, $s0, 2

add $t3, $t3, $s2 // $t3 = D + a<<2

sw $t2, 0($t2) // D[a] = b + a

addi $s0, $s0, 1 // a+=1

Exit:

1. 把下面的MIPS代码翻译成C代码。假设变量f,g,h,i和j分别赋值给寄存器$s0,$s1,$s2,$s3和$s4，数组A和数组B的基地址分别存放在$s6和$s7中。

addi $t0,$s6,4

add $t1,$s6,$0

sw $t1,0($t0)

lw $t0,0($t0)

add $s0,$t1,$t0

答：

A[1] = &A[0];

f = A[1] + A

1. 假定字变量f, g, h, i, j分别对应寄存器$s0, $s1, $s2, $s3, $s4, 并且字数组A和B的起始地址分别存放在寄存器$s6, $s7中，请分别采用MIPS汇编指令实现其功能：

a) f=g+h+B[4]; b) f=g-A[B[4]];

c) f=g+h+B[1]; d) f=A[B[g]+1];

答：

(a)

lw $t0, 16($s7) // $t0 = B[4]

add $t1, $s1, $s2 // $t1 = g + h

add $s0, $t0, $t1 // f = $t0 + $t1

(b)

lw $t0, 16($s7) // $t0 = B[4]

sll $t0, $t0, 2 // $t0 = $t0 << 2

lw $t1, 0($t0) // $t1 = A[B[4]]

sub $s0, $s1, $t1 // f = g – A[B[4]]

(c)

lw $t0, 4($s7) // $t0 = B[1]

add $t1, $s1, $s2 // $t1 = g + h

add $s0, $t0, $t1 // f = $t0 + $t1

(d)

sll $t0, $s0, 2 // $t0 = g << 2

add $t1, $t0, $s7 // $t1 = B + 4\*g

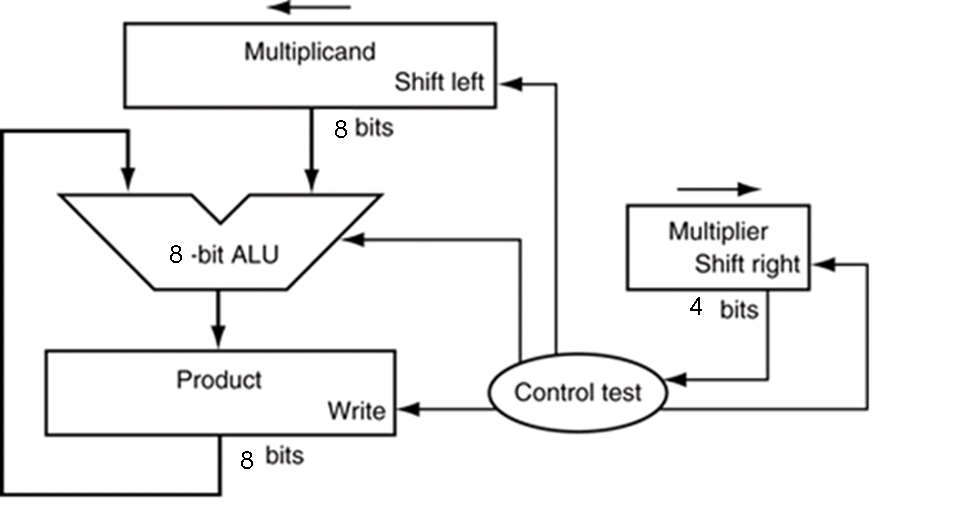
lw $t2, 0($t1) // $t2 = B[g]

sll $t2, $t2, 2 // $t2 = $t2 << 2

add $t1, $t2, $s6 // $t1 = A + B[g]\*4

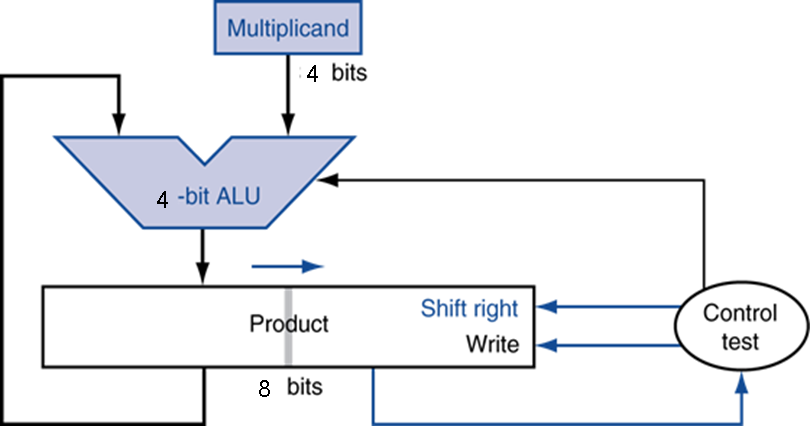
lw $s0, 4($t1) // f = A[B[g] + 1]

1. 请画出未经优化的乘法器结构图（含部件名称、位长、是否具有移位以及移位方向，部件间的连接），假设机器字长为4位，并填写无符号二进制数0101ⅹ1011的计算细节步骤。



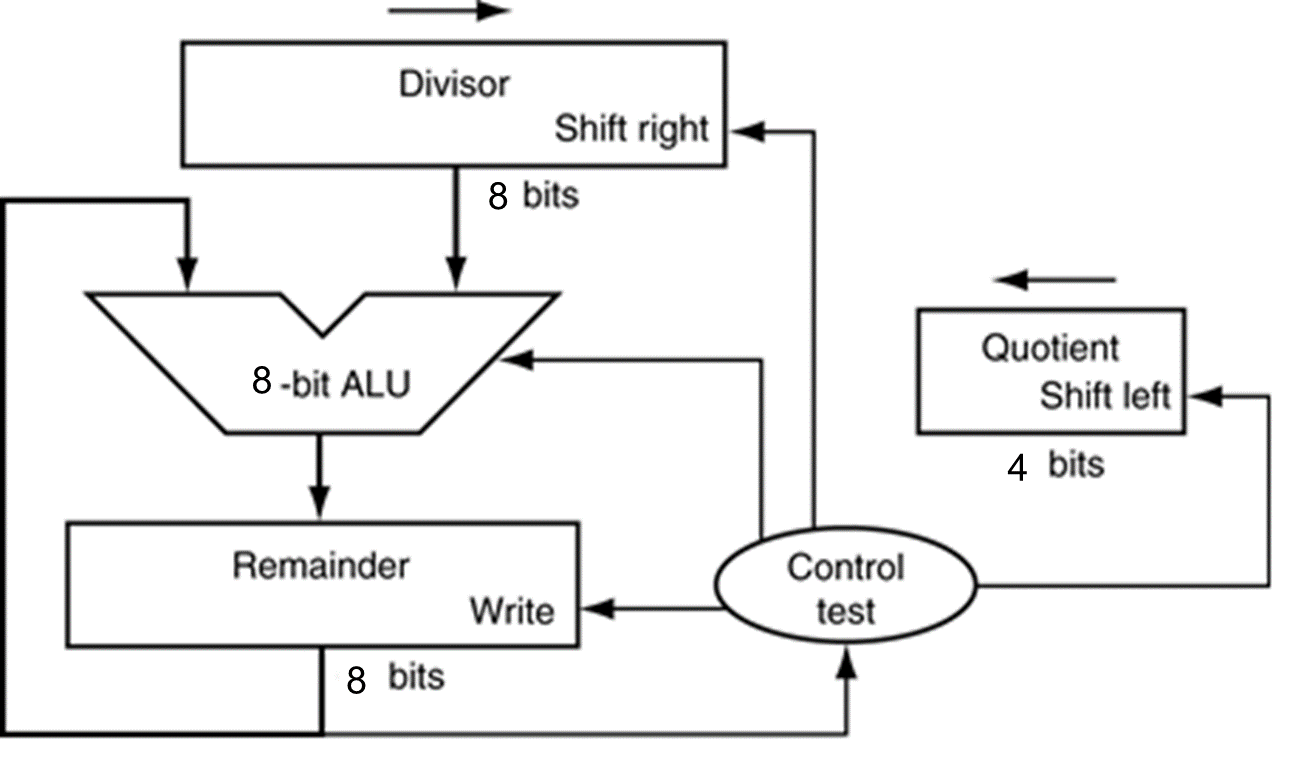
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **步骤** | **乘数寄存器** | **被乘数寄存器** | **乘积寄存器** |
| 0 | 初始值 | 0101 | 0000 1011 | 0000 0000 |
| 1 | 1a 乘积=乘积+被乘数 | 0101 | 0000 1011 | 0000 1011 |
| 2 左移被乘数 | 0101 | 0001 0110 | 0000 1011 |
| 3 右移乘数 | 0010 | 0001 0110 | 0000 1011 |
| 2 | 1 无操作 | 0010 | 0001 0110 | 0000 1011 |
| 2 左移被乘数 | 0010 | 0010 1100 | 0000 1011 |
| 3 右移乘数 | 0001 | 0010 1100 | 0000 1011 |
| 3 | 1a 乘积=乘积+被乘数 | 0001 | 0010 1100 | 0011 0111 |
| 2 左移被乘数 | 0001 | 0101 1000 | 0011 0111 |
| 3 右移乘数 | 0000 | 0101 1000 | 0011 0111 |
| 4 | 1 无操作 | 0000 | 0101 1000 | 0011 0111 |
| 2 左移被乘数 | 0000 | 1011 0000 | 0011 0111 |
| 3 右移乘数 | 0000 | 1011 0000 | 0011 0111 |

1. 请画出优化后的乘法器结构图（含部件名称、位长、是否具有移位以及移位方向，部件间的连接），假设机器字长为4位，并填写无符号二进制数0101ⅹ1011的计算细节步骤。



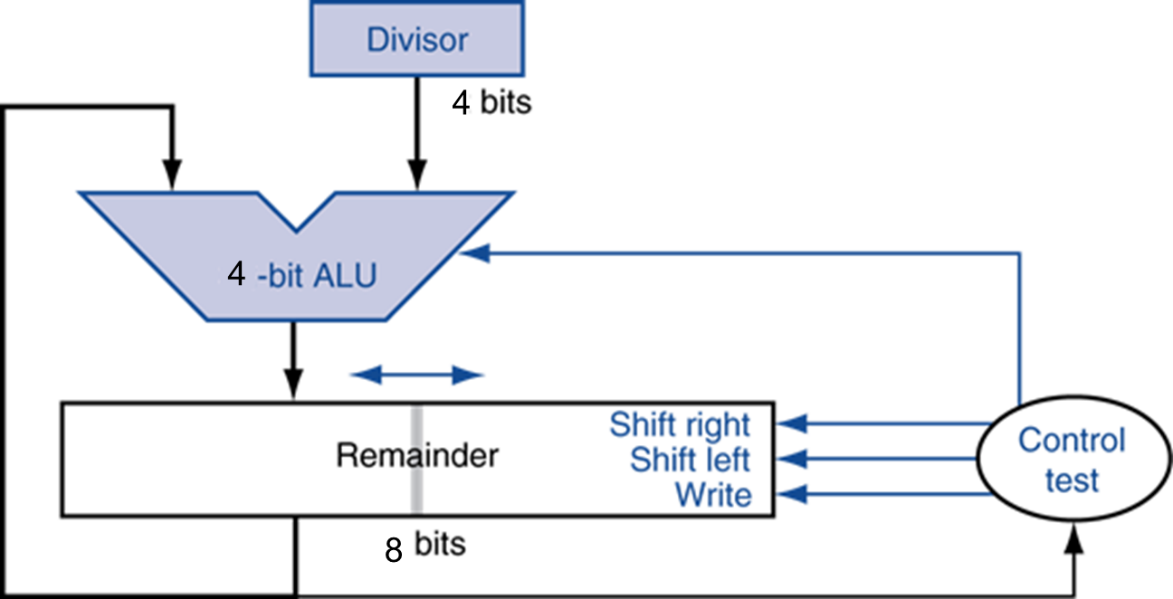
|  |  |  |  |
| --- | --- | --- | --- |
|  | **步骤** | **被乘数寄存器** | **乘积寄存器** |
| 0 | 初始值 | 1011 | 0000 0101 |
| 1 | 1a乘积=乘积（高位）+被乘数 | 1011 | 1011 0101 |
| 2 右移乘积 | 1011 | 0101 1010 |
| 2 | 1 无操作 | 1011 | 0101 1010 |
| 2右移乘积 | 1011 | 0010 1101 |
| 3 | 1a乘积=乘积（高位）+被乘数 | 1011 | 1101 1101 |
| 2 右移乘积 | 1011 | 0110 1110 |
| 4 | 1 无操作 | 1011 | 0110 1110 |
| 2右移乘积 | 1011 | 0011 0111 |

1. 请画出未优化的除法器结构图（含部件名称、位长、是否具有移位以及移位方向，部件间的连接），假设机器字长为4位，并填写无符号数7÷2的计算细节步骤。



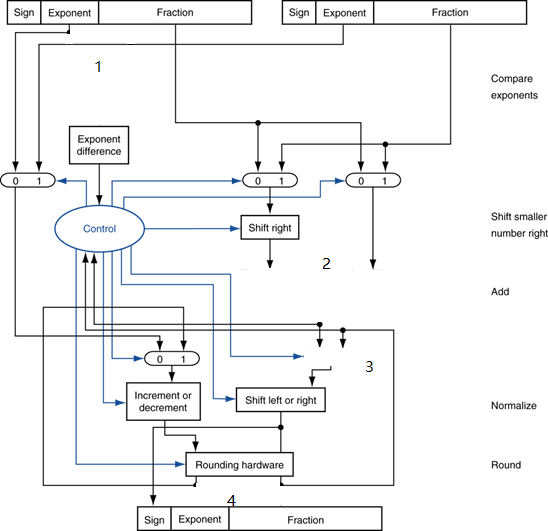
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **步骤** | **商寄存器** | **除数寄存器** | **余数寄存器** |
| 0 | 初始值 | 0000 | 0010 0000 | 0000 0111 |
| 1 | 1 余数=余数-除数 | 0000 | 0010 0000 | 1110 0111 |
| 2 余数<0，余数复原，商左移 | 0000 | 0010 0000 | 0000 0111 |
| 3 除数右移 | 0000 | 0001 0000 | 1110 0111 |
| 2 | 1 余数=余数-除数 | 0000 | 0001 0000 | 1111 0111 |
| 2 余数<0，余数复原，商左移 | 0000 | 0001 0000 | 0000 0111 |
| 3 除数右移 | 0000 | 0000 1000 | 0000 0111 |
| 3 | 1 余数=余数-除数 | 0000 | 0000 1000 | 1111 0111 |
| 2 余数<0，余数复原，商左移 | 0000 | 0000 1000 | 0000 0111 |
| 3 除数右移 | 0000 | 0000 0100 | 0000 0111 |
| 4 | 1余数=余数-除数 | 0000 | 0000 0100 | 0000 0011 |
| 2余数>0，商左移，最低位设1 | 0001 | 0000 0100 | 0000 0011 |
| 3除数右移 | 0001 | 0000 0010 | 0000 0011 |
| 5 | 1 余数=余数-除数 | 0001 | 0000 0010 | 0000 0001 |
| 2余数>0，商左移，最低位设1 | 0011 | 0000 0010 | 0000 0001 |
| 3除数右移 | 0011 | 0000 0001 | 0000 0001 |

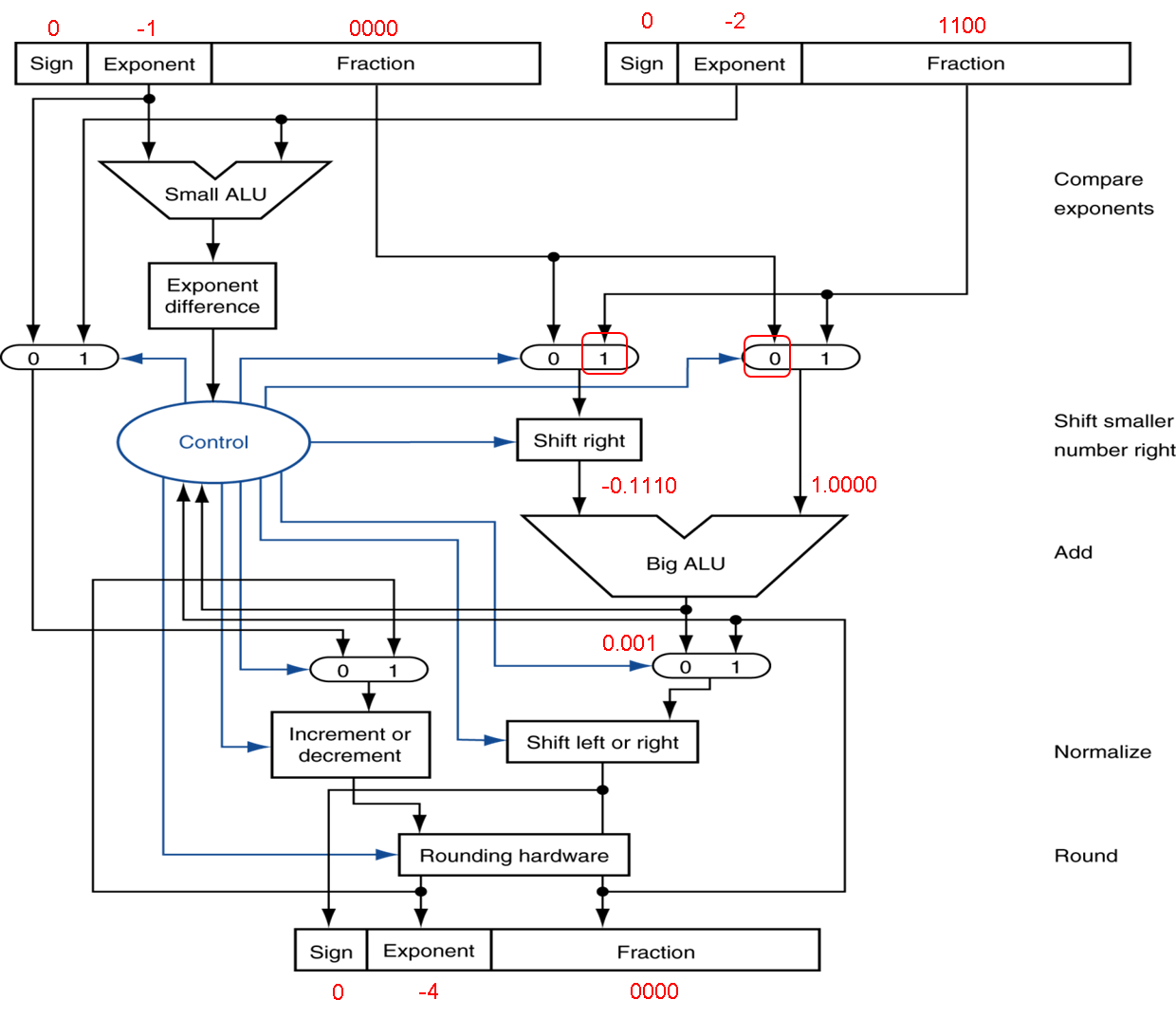
1. 请画出优化后除法器结构图（含部件名称、位长、是否具有移位以及移位方向，部件间的连接），假设机器字长为4位，并填写无符号数7÷2的计算细节步骤。



|  |  |  |  |
| --- | --- | --- | --- |
|  | **步骤** | **除数寄存器** | **余数寄存器** |
| 0 | 初始值 | 0010 | 0000 0111 |
| 1 | 1 余数左移 | 0010 | 0000 1110 |
| 2 余数（高位）=余数（高位）-除数 | 0010 | 1110 1110 |
| 3 余数<0，余数复原 | 0010 | 0000 1110 |
| 2 | 1 余数左移 | 0010 | 0001 1100 |
| 2 余数（高位）=余数（高位）-除数 | 0010 | 1111 1100 |
| 3 余数<0，余数复原 | 0010 | 0001 1100 |
| 3 | 1 余数左移 | 0010 | 0011 1000 |
| 2 余数（高位）=余数（高位）-除数 | 0010 | 0001 1000 |
| 3 余数>0，余数最低位置1 | 0010 | 0001 1001 |
| 4 | 1 余数左移 | 0010 | 0011 0010 |
| 2 余数（高位）=余数（高位）-除数 | 0010 | 0001 0010 |
| 3 余数>0，余数最低位置1 | 0010 | 0001 0011 |
| 5 | 1 |  |  |
| 2 |  |  |
| 3 |  |  |

1. 请补充完成下面的浮点加法器的结构图，将图中数字1~4的部位绘制完整。然后以下图所示的浮点加法硬件计算0.510+（-0.4375）10，精度为4位，采用IEEE 754单精度格式表示浮点数。请完成以下工作：（1）结合图中关键部件和箭头处标明相关步骤序号并在空白处说明，重点讲明数据的变换和传输，关键部件的输入和输出。（2）相关数据转换，如十进制转换为二进制等，如对阶计算过程请在空白处详细给出。





答案：

1. 由于只有4位精度，所以(0.510)10=()2; (-0.4375)10=()2
2. 对阶：
3. 尾数相加：
4. 规格化：
5. 舍入：无需舍入，