选择题：

MIPS指令长度：**32位**

半字长度：**16位（字长32位）**

字节长度：**8位**

传入参数所用的寄存器？**$a0-$a3**

Mult指令的作用？**用于执行有符号整数的乘法操作。它将两个寄存器中的值相乘，并将结果保存在两个特殊的寄存器中，即Hi（高位）寄存器和Lo（低位）寄存器。**

mult $s0, $s1 ; 将$s0和$s1中的值相乘

mfhi $hi ; 将乘法结果的高32位保存到Hi寄存器中

mflo $lo ; 将乘法结果的低32位保存到Lo寄存器中

一个cache有8个块，每个块大小4（字节），字节地址为1200的会映射到哪个块?

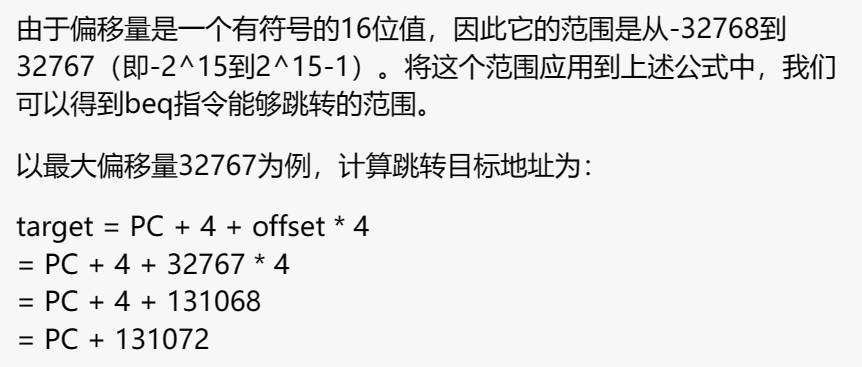
**1200/4=300 300 % 8 = 4**

PC的作用？

**程序计数器是用于存放下一条指令所在单元的地址的地方。**

PC的值=0x00000000，如果采用beq指令跳转，求能跳转的范围？

**[0x0, 0x20000]**



$t0=0xBBBBBBBB 然后andi $t0,$t0,0xFFEE答案是**0xBBAA**

MemRead的作用？**是否读内存**

简述写直达和写回的概念，写回的作用是什么？

**写直达：当数据写命中时，同时更新cache和主存**

**写回：当数据写命中时，只更新cache，而不立即更新主存。当cache被替换出时，才将被修改过的cache写回到主存。**

**写回的作用是减少对主存的写入次数，从而提高系统的性能。**

为了更快的进行编译，给出了两种代码序列，下面分别列出了两种指令序列在不同指令类型下所需的时钟周期

|  |  |  |  |
| --- | --- | --- | --- |
| \指令种类 | A | B | C |
| CPI | 1 | 2 | 3 |
| 代码序列1 | 2 | 1 | 2 |
| 代码序列2 | 4 | 2 | 1 |

1. 计算下面两种指令序列的在三种指令类型下总共需要执行的指令数**（同一行全部加起来即可）**
2. 这两种指令哪个更快？**（求CPU时间 也就是1x2+2x1+3x2=10和4x1+2x2+3x1=11 然后10小于11 所以序列1更快）**

求这两种代码序列的CPI **用CPU时间10/（2+1+2）和11/（4+2+1）**

将这段指令翻译成MIPS语言，其中ab的值存放在$s0,$s1,然后再给出两个寄存器$t0,$t1供你使用

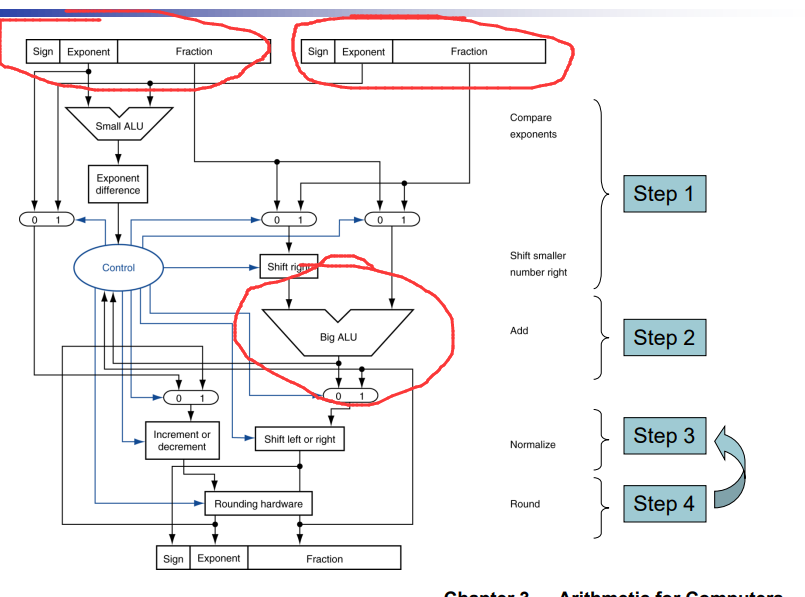
While（a<0）{

C[a]=a+b;

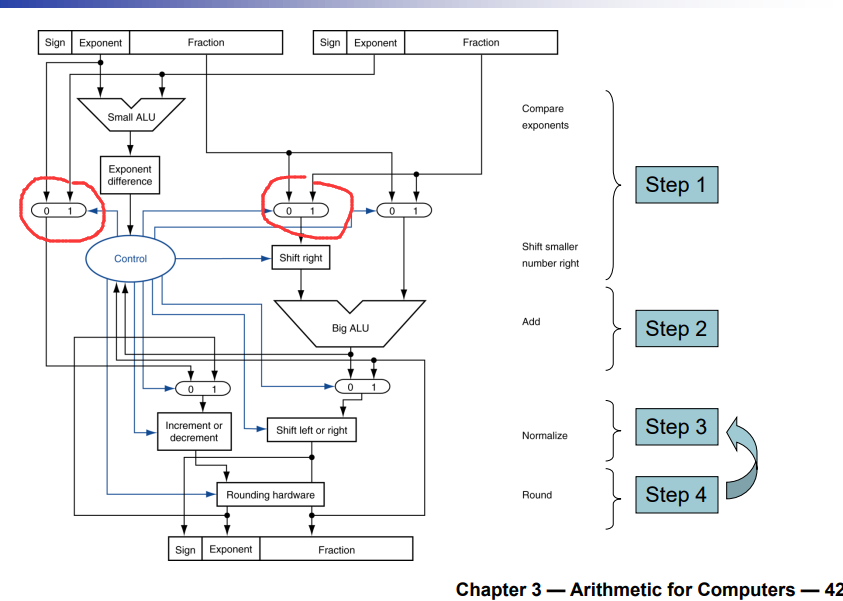
a+=1;

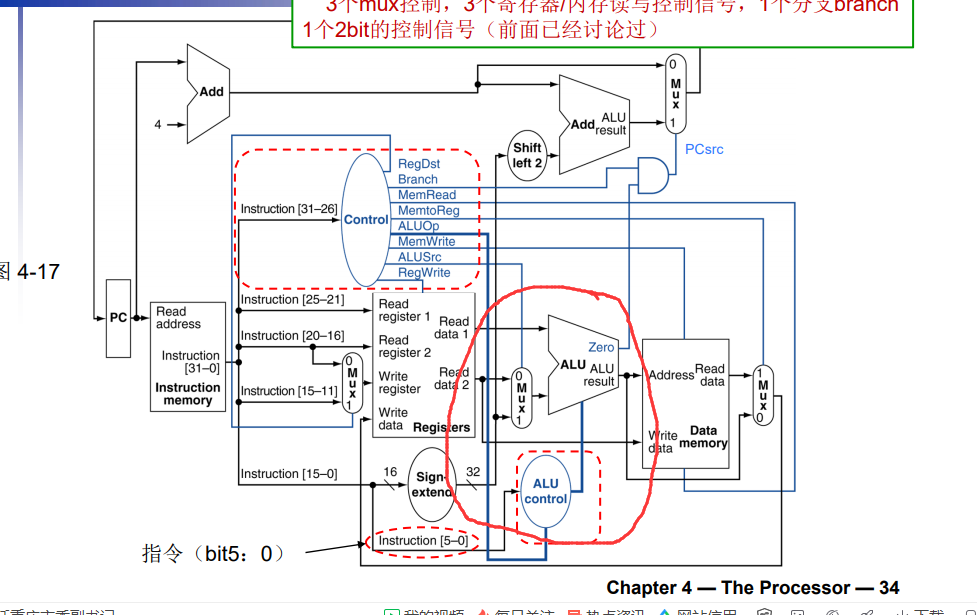
}

删去红框中的内容让你画出来



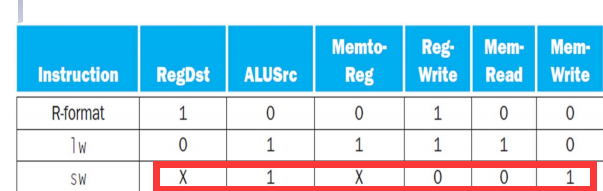
框中两个复选器的作用？



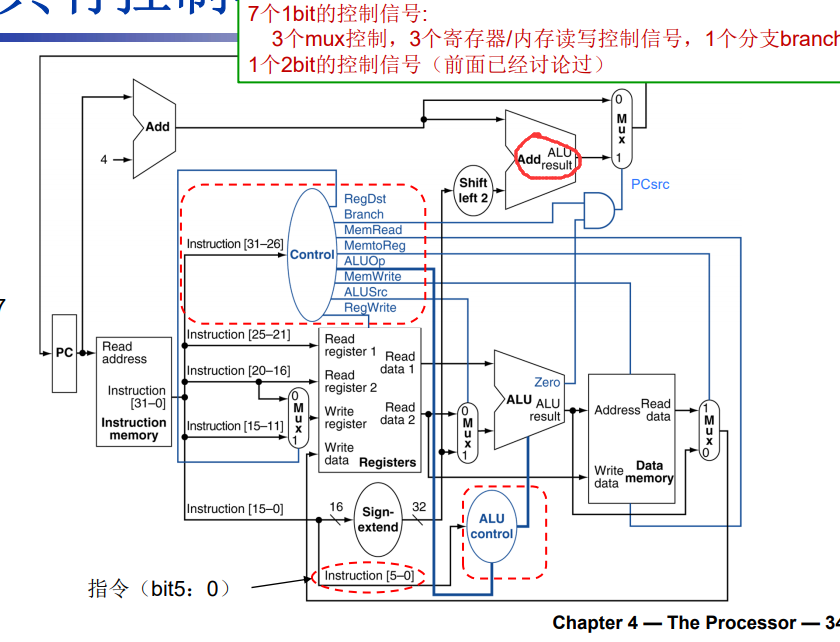


拆掉红框中这一块让你补充

在sw指令中 填空这红框的值：



如果sw指令中的立即数=100 pc的值=1000，则下图红框中计算出的值是多少？



Lw $t1,10($s0)

Add $t3,$t1,$t0

Sw $s3,20($t3)

1. 哪些指令和寄存器会发生数据冒险

**$s0：Lw $t1,10($s0) 和 Add $t3,$t1,$t0，**

**$t3：Add $t3,$t1, $t0和Sw $s3,20($t3**)

1. 不采用旁路的方法 需要在哪里加入nop指令 加入多少条?

**Lw $t1,10($s0) 和 Add $t3,$t1,$t0之间3条**

**Add $t3,$t1,$t0和Sw $s3,20($t3) 之间2条**

3.是否可以采用旁路的方法解决数据冒险？能请画出对应的逻辑判断表达式 不能请说明理由

**不可以完全解决Lw $t1,10($s0) 和 Add $t3,$t1,$t0的冒险。**

**可以解决Add $t3,$t1, $t0和Sw $s3,20($t3)的冒险。**

**If (EX/MEM.RegWrite and**

**EX/MEM.RegisterRd != 0 and**

**EX/MEM.RegisterRd=ID/EX.RegisterRs) ForwardA = 10;**

**If (EX/MEM.RegWrite and**

**EX/MEM.RegisterRd != 0 and**

**EX/MEM.RegisterRd = ID/EX.RegisterRt) ForwardB = 10;**

一个cache 4个块 每个块大小为1个字，采用直接映射和二路组相联的方法来缓存以下字地址的数据：6、14、0、14、8、14、20、8

1. 求直接映射 cache的命中率
2. 二路组相联的命中率