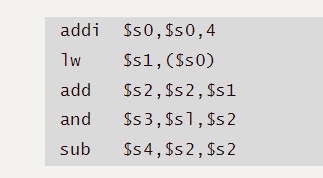
计组手写作业7

7.2选择题

1-5 CADBC 6-10 ADACB

7.8如果采用气泡流水线执行下述程序，请给出类似图7.18所示的流水线时空图。注意时空图中最后一个时钟周期第5条指令进入ID段。

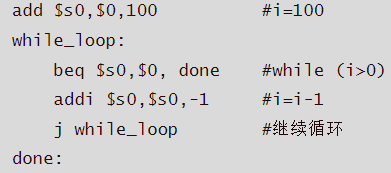


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLKs** | **取指令IF** | **译码ID** | **执行EX** | **访存MEM** | **写回WB** |
| **1** | addi |  |  |  |  |
| **2** | lw | addi |  |  |  |
| **3** | add | lw | addi |  |  |
| **4** | add | lw |  | addi |  |
| **5** | add | lw |  |  | addi |
| **6** | and | add | lw |  |  |
| **7** | and | add |  | lw |  |
| **8** | and | add |  |  | lw |
| **9** | sub | and | add |  |  |
| **10** | sub | and |  | add |  |
| **11** | sub | and |  |  | add |
| **12** |  | sub | and |  |  |

7.9如果采用重定向流水线执行7.8中程序，请给出类似图7.18所示的流水线时空图。注意时空图中最后一个时钟周期第5条指令进入ID段。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **CLKs** | **取指令IF** | **译码ID** | **执行EX** | **访存MEM** | **写回WB** |
| **1** | addi |  |  |  |  |
| **2** | lw | addi |  |  |  |
| **3** | add | lw | addi |  |  |
| **4** | and | add | lw | addi |  |
| **5** | and | add |  | lw | addi |
| **6** | sub | and | add |  |  |
| **7** |  | sub | and | add | lw |

7.10假设重定向流水线中所有分支跳转指令均在EX段执行，无分支预测、无分支延迟槽技术，尝试计算下述程序的执行周期。

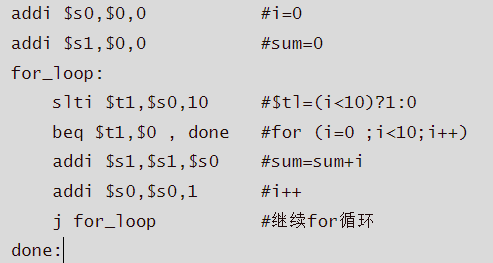


指令填满流水线需要4个时钟周期，流水线采用重定向技术，这里不存在Load-Use，所以不会因为Load-Use插入气泡，所以可以认为循环中的addi的CPI=1；beq指令跳转成功则需要清空误区的指令，会产生两个气泡，所以CPI=3，如果不成功跳转则CPI=1，因此总共需要4+1+100\*（1+1+3）+3=508个周期

7.11假设重定向流水线中所有分支跳转指令均在EX段执行，采用动态分支预测技术，beq、j指令都可以进行预测，计算最优情况下7.10中程序的执行周期。

在最优情况下第一次读取循环指令时beq和j指令无法查表得到，beq指令在其执行阶段进入BTB表中并预测不跳转，j指令在其执行阶段进入BTB并且一直预测跳转，第一次循环中因为j指令需要清空指令所以需要5个时钟周期，剩下的99次循环都预测成功，所以需要297个时钟周期，第101次进入循环时beq预测不跳转但是发生跳转，所以需要3个时钟周期，因此总共需要4+1+5+297+3=310个时钟周期

7.12假设重定向流水线中所有分支跳转指令均在EX段执行，设无分支预测、无分支延迟槽技术,试计算下述程序的执行周期。



循环一共进行10次，这10次内beq指令都不会发生跳转，所以CPI=1，j指令会成功跳转，因此其CPI=3，第11次进入循环语句时，循环跳出，所以beq的CPI=3，因此总的时钟周期为4+2+10\*7+1+3=80

7.15某16位计算机中，带符号整数用补码表示，数据cache和指令cache分离。表7.8中给出了指令系统中的部分指令格式，其中 Rs和Rd表示寄存器，mem表示存储单元地址。另外，(x)表示寄存器x或存储单元x的内容。



该计算机采用5段流水线方式执行指令，各流水段分别是取指令(IF)、译码/读寄存器(ID)、执行/计算有效地址(EX)、访问存储器(M)和结果写回寄存器( WB)，流水线采用“按序发射，按序完成"方式,没有采用转发技术处理数据相关问题,并且同一个寄存器的读和写操作不能在同一个时钟周期内进行。请回答下列问题。

(1)若int型变量x的值为-513,存放在寄存器R1中，则执行指令“SHLR1”后，R1的内容是多少?(用十六进制表示)

(2)若某个时间段中，有连续的4条指令进入流水线，在其执行过程中没有发生任何阻塞，则执行这4条指令所需的时钟周期数为多少?

(3)若高级语言程序中某赋值语句为x=a+b，x、a和 b均为 int型变量，它们的存储单元地址分别表示为[x]、[a]和[b]。该语句对应的指令序列及其在指令流水线中的执行过程如图7.32所示，则这4条指令执行过程中，I3的ID段和14的F段被阻塞的原因各是什么?



(4)若高级语言程序中某赋值语句为x=x\*2+a，x和a均为unsigned int型变量，它们的存储单元地址分别表示为[x]、[a]，则执行这条语句至少需要多少个时钟周期?要求模仿图7.32画出这条语句对应的指令序列及其在流水线中的执行过程示意图。

(1) x对应的十六进制数为FDFFH，左移一位后为FBFEH

(2) 4+4=8个时钟周期

(3)I3译码阶段需要用到R2寄存器当中的值，但此时I2指令还在执行阶段，没有完成数据的写回，因此发生数据冲突

I3阶段被阻塞在译码阶段，因此下一个时钟周期时I4也无法进入译码阶段，被阻塞在了取指阶段

(4)因为该计算机无法实现在一个时钟周期内完成同一个寄存器的读和写，因此对因同一个寄存器产生的数据冲突需要插入三个气泡，直到将数据写回后的下一个时钟周期才能进行下一条指令的译码，为了让该指令时间尽可能短，将两条LOAD指令连续执行可以减少时间冲突

I1 LOAD R1，[x]

I2 LOAD R2,[a]

I3 SHL R1

I4 ADD R1,R2

I5 STORE R1,[x]

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 时间单元 | | | | | | | | | | | | | | | |
| 指令 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| I1 | IF | ID | EX | M | WB |  |  |  |  |  |  |  |  |  |  |  |
| I2 |  | IF | ID | EX | M | WB |  |  |  |  |  |  |  |  |  |  |
| I3 |  |  | IF |  |  | ID | EX | M | WB |  |  |  |  |  |  |  |
| I4 |  |  |  |  |  | IF |  |  |  | ID | EX | M | WB |  |  |  |
| I5 |  |  |  |  |  |  |  |  |  | IF |  |  | ID | EX | M | WB |

7.16某程序中有如下循环代码段p:“for(int i= 0; i <N; i++) sum+=A[i];”。假设编译时变量sum和i分别分配在寄存器Rl和R2中，常量N在寄存器R6中，数组A的首地址在寄存器R3中。程序段p的起始地址为08048100H，对应的汇编代码和机器代码如表7.9所示。



执行上述代码的计算机M采用32位定长指令字，其中分支指令 bne采用图7.33所示格式。

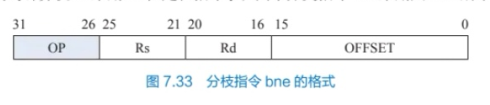


图7.33中，OP为操作码;Rs和Rd为寄存器编号;OFFSET为偏移量，用补码表示。请回答下列问题,并说明理由。

(1)M的存储器编址单位是什么?

(2)已知sll指令实现左移功能，数组A中每个元素占多少位?

(3)bne指令的OFFSET字段的值是多少?已知 bne指令采用相对寻址方式，当前PC的内容为bne指令地址，通过分析表7.9中的指令地址和 bne指令内容，推断出 bne指令的转移目标地址计算公式。

(4)若M采用“按序发射、按序完成”的5级指令流水线——IF(取指令)、ID(译码及取数)、EX(执行)、MEM(访存)、WB(写回寄存器)，且硬件不采取任何转发措施，分支指令的执行均引起3个时钟周期的阻塞,则p中哪些指令的执行会因数据相关而发生流水线阻塞?哪条指令的执行会发生控制冒险?为什么指令1的执行不会因为与指令5数据相关而发生阻塞?

(1)一条指令的长度是32位，两条指令之间在地址上差4，因此说明一个地址单元为8位，即按照字节编址

(2)由R2里的内容左移两位可知数组的一个元素占据两个存储位置，因此每个元素占据16位

(3)该程序中bne指令的机器代码为1446FFFAH，其OFFSET字段对应的值为FA，换算为十进制为-6，从bne所在指令跳回loop对应的位置需要地址-18H，因此可以直到对应的bne目标地址计算公式为(PC)+OFFSET<<2+4

(4)指令2、3、4、6会和它们的前一条指令发生数据冲突；bne指令的执行会造成控制冒险；如果发生跳转，指令6会插入两个气泡，所以需要3个时钟周期，指令5从执行到数据写回只需要2个时钟周期，所以当执行为指令6后如果跳转到指令1已经由足够的时间让指令5完成数据写回