**一、简述题（30分，每小题6分）：**

1. 数据存储是一般都是字节的整数倍，例如32位数据是4个字节，这32位的数据一般可以有2种顺序存放，请说明是哪2种存放顺序。若数据12345678H(16进制)存放在100、101、102、103这4个存储单元中，按照这2种存放方式，各存储单元的数据值各是多少？

**答案：**2种存放方式为**大端存储**（最左边字节是字地址，高位数据放入低地址）和**小端存储**（最右边字节放入字地址，低位数据放入低地址），所以，大端存储：各个单元的数值为**100单元值12、101单元值34、102单元值56、103单元值78**；小端存储**100单元值78；101单元值56、102单元值34、103单元值12**。

1. MIPS指令集的三项设计原则是什么？举例说明这些原则是如何应用到具体指令中的？

**答案：**MIPS指令集的三项设计原则是：**1）简单源于规整**，例如：add $1 $2 $3,每个指令有且仅有三个操作数；**2）越小越快**，例如：指令集中只要32个寄存器；**3）优秀的设计需要好的折中**，例如：保持指令长度相同，但是不同类型的指令采用不同的指令格式，add $1, $2, $3、lw $1, 50($2) 。

1. 旁路是如何解决数据冒险的？是否能解决所有的数据冒险？并举例说明。

**答案：**旁路也称为 “前推”，是一种解决流水线数据冒险的方法，从内部寄存器二非程序员可见的寄存器或存储器中**提前取出数据**。例如add **$0**,$1,$2;sub $3,**$0**,$2;指令sub不需要等add执行结束，ALU 一旦生成加法的运算结果，sub指令就**从部件内部（旁路）直接获得**这个结果参与sub的运算，而不是等待add指令执行结束。

1. 直接相联映射、组相联映射、全相联映射的共同点是什么？它们之间的关系是什么？

**答案：**直接相联映射、组相联映射、全相联映射的共同点是**主存和cache都是分成同样大小的存储区块，组相联映射是直接相联映射和全相联映射的折中方案，若cache只有一组则为全相联映射，若一组中只有一块这为直接相联映射。**

1. **答案：**下表是TLB，页表以及cache访问中可能的组合，完成表第四栏的内容填写

|  |  |  |  |
| --- | --- | --- | --- |
| TLB | 页表 | Cache | 这种情况能发生吗？，  如果可能，在什么情况下发生？ |
| 命中 | 命中 | 缺失 | 可能，但TLB命中不可能检查页表 |
| 缺失 | 命中 | 命中 | TLB缺失，但在页表中找到表项；重试后在cache中找到数据 |
| 缺失 | 命中 | 缺失 | TLB缺失，随后发生却页；重试后在cache中找不到数据 |
| 缺失 | 缺失 | 缺失 | TLB缺失，但在页表中找到表项；重试后在cache中未找到数据 |
| 命中 | 缺失 | 缺失 | 不可能：如果页不在主存中，TLB中没有此转换 |
| 命中 | 缺失 | 命中 | 不可能：如果页不在主存中，TLB中没有此转换 |
| 缺失 | 缺失 | 命中 | 不可能：如果页不在主存中，数据不允许在cache中出现 |

**二、计算题（20分，每题10分）：**

**1.（10分）**同一个指令集 体系结构有两种不同的实现方式，有A、B、C、D，4类指令。每种实现方式的时钟频率和CPI由下表给定。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 时钟频率 | CPI(A类) | CPI(B类) | CPI(C类) | CPI(D类) |
| 体系结构P1 | 1.5GHz | 1 | 2 | 3 | 4 |
| 体系结构P2 | 2 GHz | 2 | 2 | 2 | 2 |

假定一个程序有106条指令，其中各类指令的比例为A类占10%、B类占20%、C类占50%、D类占20%，问哪种实现方式更快（给出理由）？

答案：P2实现方式更快。

A类指令为10610%=105 条指令；B类指令为10620%=2105 条指令；

C类指令为10650%=5105 条指令；D类指令为10620%=2 105 条指令；

处理器运行程序的时间 指令数CPI/时钟频率

使用P1处理器各类指令的运行时间为:

A类指令 105 /1.5 GHz=0.66秒；B类指令 2105/1.5 GHz=2.66秒；

C类指令 505/1.5 GHz=10秒； D类指令 2105/1.5 GHz=5.33秒；

P1 总时间18.65秒。同理求得：P2 总时间11秒。所以，P2实现方式更快。

**2.(10分)**

内存按照字节编址，MIPS程序段如X下（十进制表示），若将此程序段编译后调入起始地址为2000内存单元中，请给出Start、Exit、Target在对应的内存位置的数值。如果寄存器中的初值等于其编号值，存储器单元中的数值等于其存储单元地址值，那么执行完该程序段后，哪些寄存器中的值发生变化，变化后的寄存器中的数值是多少？

Start: sub $10, $4, $7  
beq $1, $3, Target  
j Exit

Target: and $12, $2, $5;

or $13, $2, $6  
 add $14, $4, $2  
 slt $15, $6, $7  
 or $13,$6,$2  
 add $14,$2,$2  
 sw $15,102($2)

Exit: lw $4, 56($8)

**答案：**PC高位是0，所以，只考虑地位数值即可。**Start=2000、Exit=2040÷4=510、Target= 1(**beq $1, $3, offset；目标地址=**PC+4+offset×4，即offset=1)；只有$10和$4中的值发生了变化，其值为$10=4-7=-3；($8)+56=64地址单元的数值，依题意，$4=64。**

Start 2000： sub $10, $4, $7

2004： beq $1, $3, 1

2008： j 510

Target2012: and $12, $2, $5;

2016： or $13, $2, $6  
 2020： add $14, $4, $2  
 2024： slt $15, $6, $7  
 2028： or $13,$6,$2  
 2032： add $14,$2,$2  
 2036： sw $15,102($2)

Exit 2040: lw $4, 56($8)

**三、综合与设计题（共50分）：**

**1.（20分）**某计算机存储器按字节编址，虚拟（逻辑）地址空间大小为4GB，主存（物理）地址空间大小为256MB，页面大小为64KB。Cache采用二路组相联直映射，共16块；主存与Cache之间交换的块大小为256B。该机配置了全相联的TLB，该TLB共可存放8个页表项。在某时刻，该计算机系统TLB、页表和Cache内容如图所示：

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| （a）页表内容 | | | | (b)cache内容 | | | | | | | （c）TLB内容 | | |
| 虚页号 | 有效位 | 物理页号 | 。。 | 组号 | 有效位 | 标记 | ...... | 有效位 | 标记 | …… | 有效位 | 标记 | 物理页号 |
| 0 | 1 | 005 | …… | 0 | 1 | 10200 | …… | 1 | 10D00 | …… | 0 | - | - |
| 1 | 1 | 01B | …… | 1 | 0 | --- | …… | 0 | --- | …… | 1 | 0011 | 155 |
| 2 | 1 | 115 | …… | 2 | 1 | 004C0 | …… | 1 | 014C0 | …… | 0 | - | - |
| 3 | 1 | 102 | …… | 3 | 1 | 101D2 | …… | 1 | 00250 | …… | 1 | 0120 | 1FD |
| 4 | 0 | - | …… | 4 | 1 | 00640 | …… | 1 | 04640 | …… | 1 | 0132 | 2D1 |
| 5 | 1 | 128 | …… | 5 | 1 | 0141A | …… | 1 | 11400 | …… | 0 | - |  |
| 6 | 0 | - | …… | 6 | 0 | --- | …… | 0 | --- | …… | 1 | 008D | 7EF |
| 7 | 1 | 32C | …… | 7 | 1 | 0272B | …… | 1 | 0174B | …… | 0 | - |  |

当CPU给出虚存地址0005C3EFH时，详细分析CPU取得数据的过程，并给出CPU取得数据后，TLB表和Cache表内容。

**参考答案**

**第一步，根据CPU给出的虚存地址访问TLB**

按照虚地址格式划分为：

|  |  |
| --- | --- |
| 虚页号（16位） | 页内地址（16位） |
| 0000 0000 0000 0**101** | 1100 0011 1110 1111 |

查找TLB所有项，不命中，访问页表，得到实地址 128C3EF，并将TLB更新为：

|  |  |  |
| --- | --- | --- |
| 有效位 | 标记 | 物理页号 |
| 0 | - | - |
| 1 | 0011 | 155 |
| **1** | **0005** | **128** |
| 1 | 0120 | 1FD |
| 1 | 0132 | 2D1 |
| 0 | - |  |
| 1 | 008D | 7EF |
| 0 | - |  |

**第二步：实地址128C3EF划分为：**

|  |  |  |
| --- | --- | --- |
| 标记（17位） | 组号（3位） | 块内位移（8位） |
| 0 0010 0101 0001 1000 | 011 | 1110 1111 |

查找Cache，不命中，访问主存，调块，放在Cache第三组第一个位置，并将数据送往CPU。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 组号 | 有效位 | 标记 | ...... | 有效位 | 标记 | …… |
| 0 | 1 | 10200 | …… | 1 | 10D00 | …… |
| 1 | 0 | --- | …… | 0 | --- | …… |
| 2 | 1 | 004C0 | …… | 1 | 014C0 | …… |
| 3 | **1** | **02518** | **……** | 1 | 00250 | …… |
| 4 | 1 | 00640 | …… | 1 | 04640 | …… |
| 5 | 1 | 0141A | …… | 1 | 11400 | …… |
| 6 | 0 | --- | …… | 0 | --- | …… |
| 7 | 1 | 0272B | …… | 1 | 0174B | …… |
|  |  |  | （b）Cache内容 | |  |  |

**2.（15分）设计流水线，**设指令流水线有取指（IF）、译码并读寄存器(ID)、计算（EX）、存储器读写和写回（MEMWB）4个阶段，假设各个阶段完成的时间依次为200ps、190ps、200ps,380ps。现有20条指令连续流入此流水线，求此流水线的实际吞吐率和加速比。 若可以重新设计流水线，应该怎样改进设计？改进后各段的完成时间是多少？20条指令连续流入改进后的流水线实际吞吐率和加速比是多少？

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| MEMWB |  |  |  | 1(380ps) | 2(380ps) | 3 | 4 | 5 | … | 20 |  |
| EXE |  |  | 1(200ps) | 2(200ps) | 3 | 4 | 5 | … | 20 |  |  |
| ID |  | 1(190ps) | 2(190ps) | 3(190ps) | 4 | 5 | … | 20 |  |  |  |
| IF | 1 | 2(200ps) | 3(200ps) | 4(200ps) | 5 | … | 20 |  |  |  |  |
|  | 200ps | 200ps | 200ps | 380ps | 380ps | … | 380ps | 380ps | 380ps | 380ps |  |

流水线的实际吞吐率=20÷(200×3+380×20) ×109=**2.4×106**;

**加速比**=[20×（200+190+200+380）]÷(200×3+380×20)=**2.37**

**应该怎样改进设计**:由上述情况可以看出，流水线的吞吐率取决于最大段的时间，因此，对MEMWB(380ps)段进行改进，将分成MEM段和WB段，即流水线分成5段，每段200ps。

**改进后各段的完成时间是多少**：每段200ps；

则20条指令流入流水线的情况如下：

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| WB |  |  |  |  | 1 | 2 | 3 | 4 | 5 | … | 20 |
| MEM |  |  |  | 1 | 2 | 3 | 4 | 5 | … | 20 |  |
| EXE |  |  | 1 | 2 | 3 | 4 | 5 | … | 20 |  |  |
| ID |  | 1 | 2 | 3 | 4 | 5 | … | 20 |  |  |  |
| IF | 1 | 2 | 3 | 4 | 5 | … | 20 |  |  |  |  |
|  | 200ps | … |  |  |  |  |  |  |  |  | 200ps |

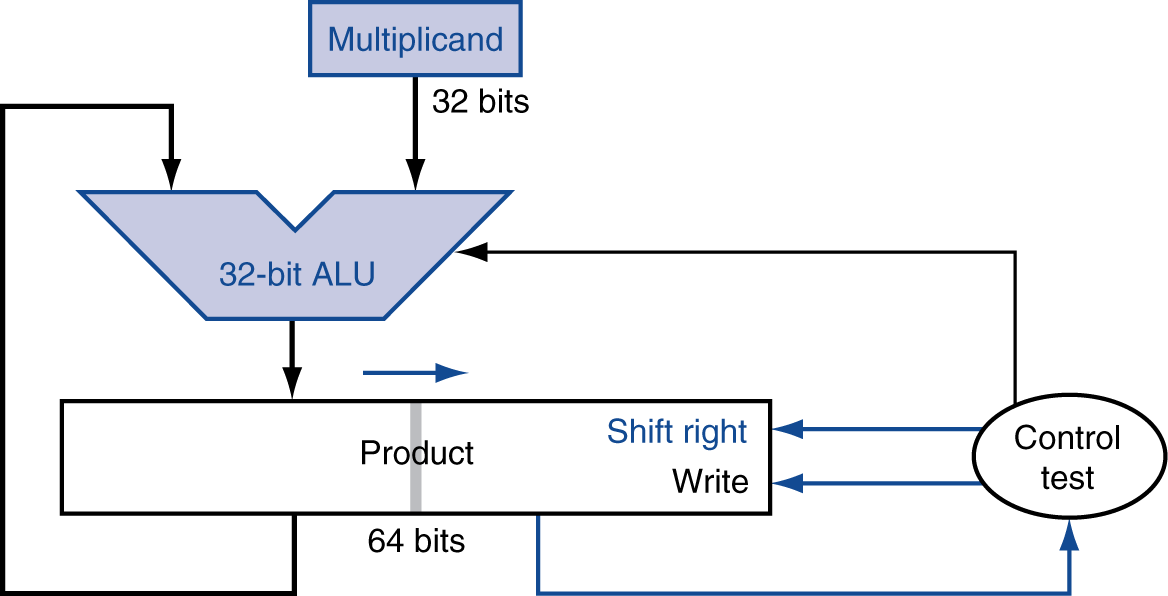
**流水线的实际吞吐率**=20÷(200×23) ×109=**4.3×106**;

**加速比**=[20×（200+190+200+380）]÷(200×23)=**4.22**

**3.（15分）**请设计可以快速实现9×5的32位乘法器，并写出详细计算步骤（写出4位二进制乘法即可）。

**解答：**

快速乘法器如下图所示：



已知被乘数为1001，乘积的低位是乘数0101，初始值为0000 0101.

详细步骤如下：

|  |  |  |  |
| --- | --- | --- | --- |
| 迭代 | 步骤 | 被乘数 | 积product |
| 0 | 初始值 | 1001 | **0000** 0101 |
| 1 | 1:最低位为1，积=积高位+被乘数 | 1001 | **1001 0101** |
| 2：右移积 | 1001 | **0100 1010** |
| 2 | 1:最低位为0，无操作 | 1001 | **0100 1010** |
| 2：右移积 | 1001 | **0010 0101** |
| 3 | 1:最低位为1，积=积高位+被乘数 | 1001 | **1011 0101** |
| 2：右移积 | 1001 | **0101 1010** |
| 4 | 1：0→无操作 | 1001 | **0101 1010** |
| 2：右移积 | 1001 | **0010 1101** |