

- 简答题

- 指令流水线的段数是不是越多性能越好？为什么？
  - 不是。
  - 流水线深度受限于流水线的延迟和额外开销。额外开销包括流水寄存器的延迟和时钟扭曲。
  - 采用流水技术，可以提高指令执行的吞吐率，从而提高程序的执行速度，但并不能减少单条指令的执行时间。由于额外开销的存在，反而会导致单条指令执行时间的增加。
- 请简述牺牲缓存（Victim Cache）的工作原理，并解释其有效性。
  - 在Cache和其下一级存储器的数据通路上增设一个全相联的小Cache，Victim Cache中存放因冲突而被替换出去的那些块。每当发生不命中时，在访问下一级存储器之前，先检查Victim Cache中是否有所需的块，若有，就将其与Cache中的某块替换。
  - 有效性：在Cache中因冲突而被替换的块很可能是最常被访问的块。
- 请简述 Tomasulo 算法与推测执行中保留站的主要功能?(6 分)推测执行中 ROB 的主要功能是什么?(4 分)
  - 保留站的主要功能是保存还没有处理完的指令、包括其操作数；此外，保留站还有一个寄存器换名的功能，它可以消除名相关来提升指令处理的性能；
  - ROB 首先得负责 tomasulo 算法中保留站的换名功能；此外，它还负责对处理完还没有提交的指令进行重排序，来确保按需提交，使得错误推测情况下执行的指令的计算结果不会被提交。
- 相比较于 RAID-3，RAID-4的优点是什么？RAID-5的优点是什么？
  - RAID-4 于 RAID-3 一样将数据条块化分布于不同的磁盘上，但条块单位为块或记录，采用粗粒度的磁盘阵列，可以使磁盘阵列并行进行多个数据的磁盘操作。
  - RAID-5 是块交叉分布奇偶校验磁盘阵列，RAID 5不对存储的数据进行备份，而是把数据和相对应的奇偶校验信息存储到组成RAID5的各个磁盘上。除了能和RAID3一样快处理大规模访问，跟RAID4一样处理小规模读操作，还能比它们更快处理小规模写操作。

- 综合应用题

1. 对于没有旁路机制（Data Forwarding）的 MIPS 5 阶段流水线，找出下面代码中的数据相关，并将相关的情况（相关**指令对**的编号以及相关的类型）列出来。（8 分）

编号	指令
1.	add \$3, \$1, \$2
2.	lw \$1, 0(\$4)
3.	and \$5, \$3, \$4
4.	and \$6, \$1, \$2
5.	or \$1, \$3, \$6
6.	sw \$1, 4(\$4)
7.	lw \$2, 4(\$4)
8.	sub \$3, \$5, \$6

- - 数据相关
    - 1、5、6、7互相数据相关
    - 1、3、8数据相关
  - 名相关

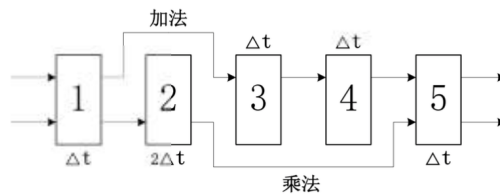
- 反相关
  - 1-2 1-5 1-7
  - 3-8
  - 4-5 4-7
  - 5-8
- 输出相关
  - 1-8
  - 2-5

○ Amdal定律

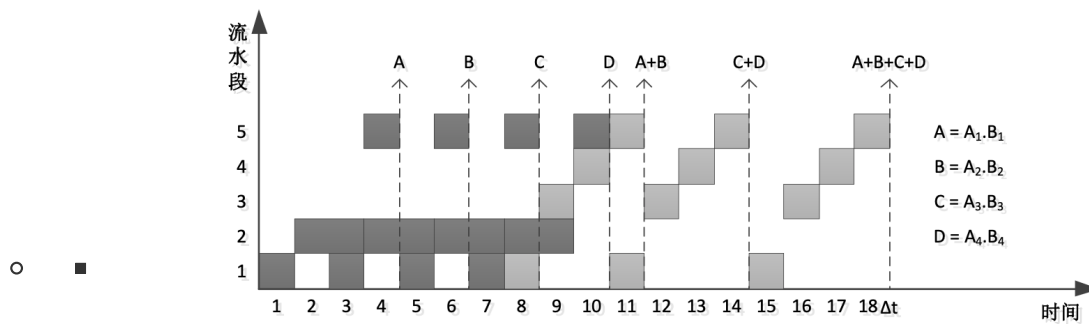
- 请用自己的语言阐述下 Amdal 定律
  - 加快某部件执行速度所能获得的系统加速比，受限于该部件的执行时间占系统总执行时间的百分比
- 一个程序的 75% 的运行时间花费在乘法运算上面。如果将乘法单元加速 3 倍，可以获得的总的加速比为多少？
  - $1 / (0.25 + 0.75/3) = 2$
- 若乘法单元可以获得无限的加速比，请问该程序能获得的最大加速比为多少
  - 4

有一条动态多功能流水线由 5 段组成（如下图所示），加法用 1、3、4、5 段，乘法用 1、2、5 段，第 2 段的时间为  $2\Delta t$ ，其余各段时间均为  $\Delta t$ ，而且流水线的输出可以直接返回输入端或

暂存于相应的流水寄存器中。若在该流水线上计算  $\sum_{i=1}^4 (A_i \cdot B_i)$ ，试计算其吞吐率、加速比和效率。（10 分）



首先,应选择适合于流水线工作的算法。对于本题,应先计算  $A_1.B_1$ 、 $A_2.B_2$ 、 $A_3.B_3$  和  $A_4.B_4$ ;  
再计算  $A_1.B_1+A_2.B_2$  和  $A_3.B_3+A_4.B_4$ ; 最后求总的结果。



根据上面的时空图,可得:

$$TP = \frac{7}{18\Delta t}$$

处理上述计算,非流水方式的时间为:  $4*4 + 3*4 = 28\Delta t$

所以,  $S = 28/18 \approx 1.56$

$$E = (4*4 + 3*4) / (5*18) \approx 0.31$$

假设你是一个工程师,目前有一个流水线处理器,其时钟周期长度为 10ns。在该处理器上,你最喜爱的基准程序的平均 CPI 为 1.6。此外,你知道该基准程序中 10% 的指令为分支指令,且该处理器中分支预测的精度为 90%。每个错误的分支预测,会造成处理器暂停 2 个时钟周期。现在你想通过增加流水线的深度使得时钟周期长度变为 9ns,但该优化措施会使得错误的分支预测的开销变为 7 个时钟周期。假定其它方面不受影响,请回答如下问题:在新的处理器中你喜爱的基准程序能够获得的 CPI 是多少? (4 分) 对于该基准程序,新的处理器与旧的相比哪个更好? (2 分)

- ■ CPU时间1 = IC \* 1.6 \* 10
- CPU时间2 = IC \* (1.6 + 0.1 \* 0.1 \* 5) \* 9

对于 CPU 中的缓存,请回答:当采取下面的改变时,强制性失效以及冲突失效分别会发生什么样的变化?并给出原因。

- (1) 将相联度增加一倍,缓存的容量和块大小不变; (3 分)
- (2) 将缓存块大小减半,缓存的相联度和缓存中组的数量不变; (3 分)
- (3) 将缓存中组的数量增加一倍,缓存容量和缓存块大小不变。 (3 分)

- ■ (1) 强制性失效不会改变,根据空间局部性原理,强制性失效只与Cache块大小有关,Cache块大小没变化,所以强制性失效不会改变;冲突失效会降低,因为冲突不命中是由于太多块映射到同一组导致。
- (2) 强制性失效增加,冲突失效增加
- (3) 强制性失效不变,冲突失效增加

一个具有一层缓存系统并支持虚拟存储器（virtual memory）的 MIPS 机器的配置说明如下：

- 1MB 物理地址空间
- 4GB 虚拟地址空间
- 4KB 页面大小
- 16KB 8路组相联缓存，写直达，LRU替换算法
- 1KB 缓存块大小
- 2个条目的TLB, LRU替换算法

假设下面的代码在该系统上单独运行，且进程上下文切换被关闭。

```
#define NUM_INTS 8192
int *A = (int *)malloc(NUM_INTS * sizeof(int)); // malloc 返回地址: 0x100000
int i, total = 0;
for(i = 0; i < NUM_INTS; i += 128) A[i] = i;
for(i = 0; i < NUM_INTS; i += 128) total += A[i]; // 目标代码
```

请根据上面的假设回答如下问题：

- (1) 假定采取字节寻址，请给出访问缓存的地址划分方案？Tag:Index:Offset （3分）
- (2) 假定采用字节寻址，请给出虚拟存储器的地址划分方案？VPN: Page Offset （3分）

对于下面三个问题，仅考虑标注 “目标代码” 的行：

- (3) 计算缓存的命中率 （3分）
- (4) 计算TLB的命中率 （3分）
- (5) 计算页表的命中率 （3分）

- ■ (1) 9 1 10
- (2) 20 12
- (3) 50%
- (4) 7/8
- (5) 100%