

CA_Assignment3

Assign	
Property	
tag	homework
姓名	周鹏宇
学号	2019K8009929039

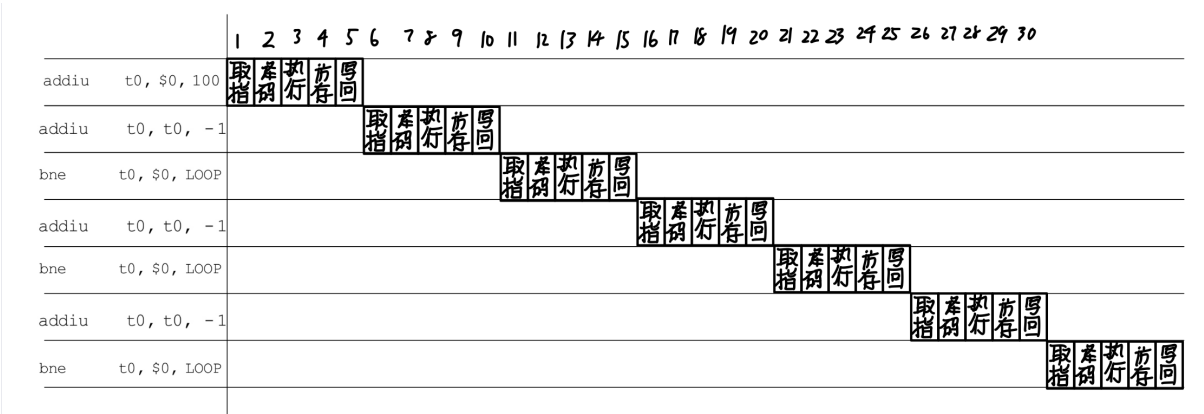
9.1

请给出下列程序在多周期处理器（如图 9.4 所示）上执行所需要的时钟周期数，并给出前三次循环执行的时空图。

```
addi.w t0, zero, 100
LOOP:
    addi.w t0, t0, -1
    bnez t0, LOOP
```

周期数为：

$$5 \times (1 + 2 \times 100) = 1005$$

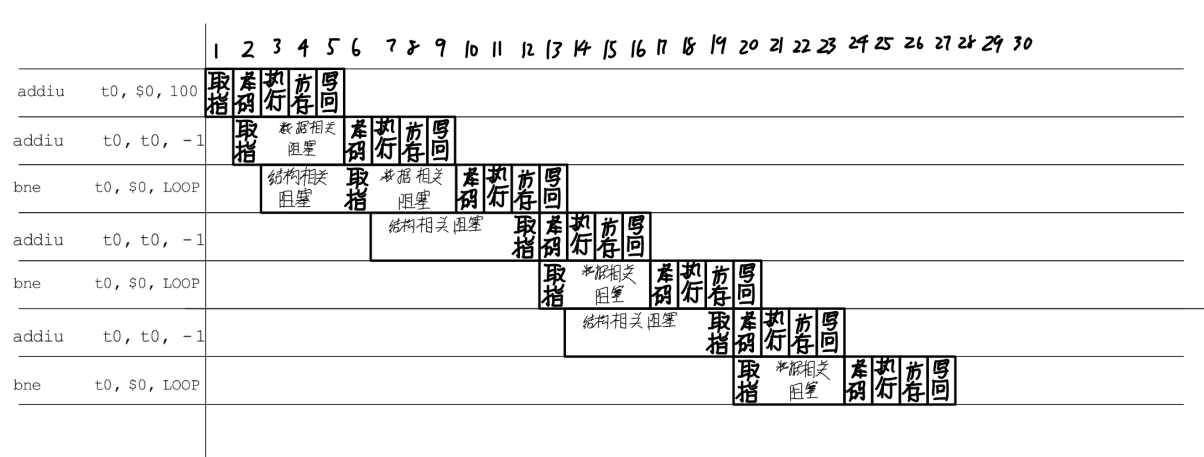


9.2

请给出题 1 中的程序在单发射 5 级静态流水线处理器（如图 9.6 所示）上执行所需要的时钟周期数，并给出前三次循环执行的流水线时空图。

考虑用阻塞的方式处理写后读和结构相关，以及当完成访存阶段后方能进入译码流水线级，则周期数为：

$$7 \times 99 + 13 = 706$$

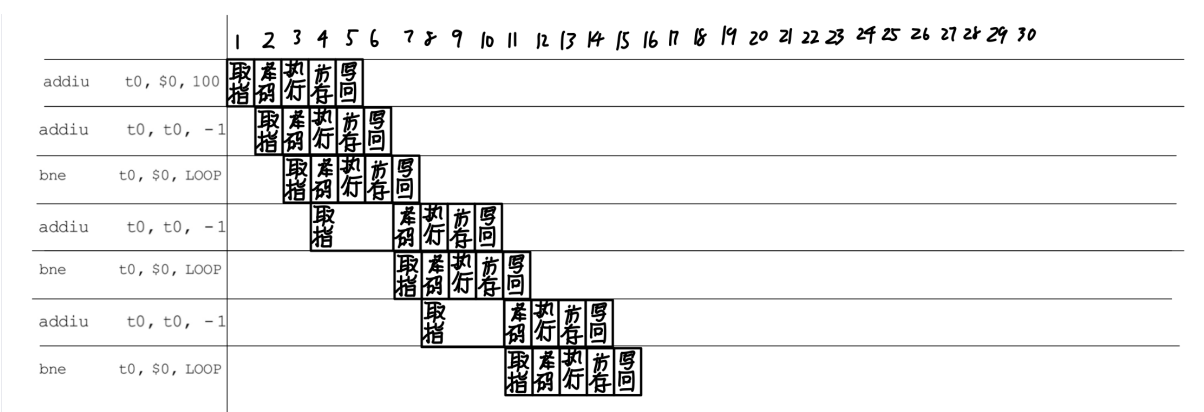


9.3

请给出题 1 中的程序在包含前递机制的单发射 5 级静态流水线处理器（如图 9.13 所示）上执行所需要的时钟周期数，并给出前三次循环执行的流水线时空图。

周期数为

$$4 \times 99 + 7 = 403$$



9.4

请在图 9.13 的基础上添加必要的逻辑，使其能够实现精确异常的功能。画出修改后的处理器结构图，并进行解释

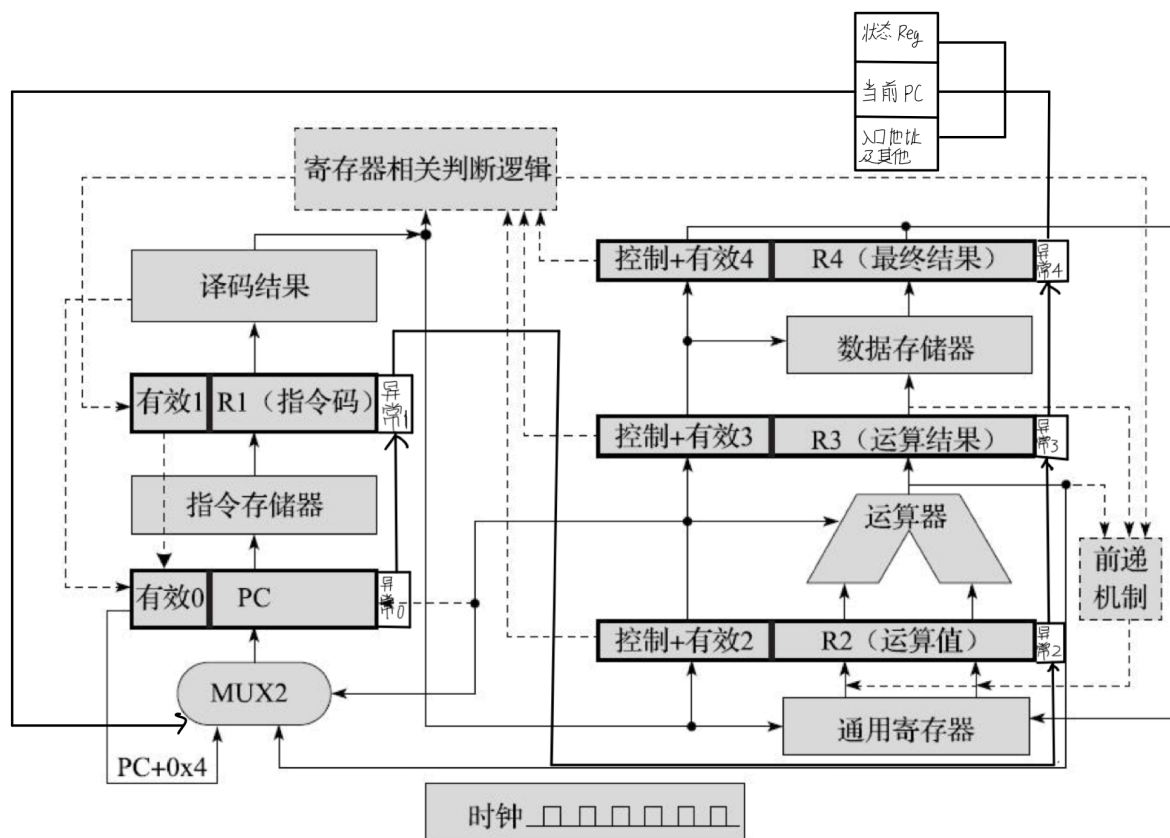


图9.13 处理指令相关的流水线结构图

- 在每个流水级都设置一个异常寄存器，用以记录可能发生的异常并保存相关信息（PC，状态等）
- 每一级流水级的异常状态都可以向下一级传递，并最终传递到一个针对异常信息的解码器（从整个信息中解码出当前PC，状态寄存器等信息）
- 在取指级增加对异常处理程序入口地址的数据选择逻辑

9.5

请给出题 1 中的程序在包含前递机制的双发射 5 级静态流水线处理器（如图 9.16 所示）上执行所需要的时钟周期数，并给出前三次循环执行的流水线时空图。

周期数为：

$$3 \times 99 + 7 = 304$$

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
addiu	t0, \$0, 100	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
addiu	t0, t0, -1	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
bne	t0, \$0, LOOP	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
addiu	t0, t0, -1					取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
bne	t0, \$0, LOOP							取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
addiu	t0, t0, -1									取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码
bne	t0, \$0, LOOP											取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码	取指	译码

9.6

请问数据相关分为哪几种?静态流水线处理器是如何解决这几种相关的?采用寄存器重命名的动态流水线是如何解决这几种相关的?

- 数据相关分为：写后读（RAW）相关（也称真相关），读后写（WAR）相关和写后写（WAW）相关
- 在简单的静态流水线中，只有RAW相关会引起流水线冲突，后两者则不会
- 解决方法为
 - 阻塞：将被阻塞流水级所在的寄存器保持原值不变，同时向被阻塞流水级的下一级流水级输入指令无效信号，用流水线空泡填充（会引起流水线执行效率的降低）
 - 前递：在流水线中读取指令源操作数的地方通过多路选择器直接把前面指令的运算结果作为后面指令的输入
- 寄存器重命名的动态流水线解决方法：
 - 写后写、读后写也会引发数据相关
 - 采用动态调度的方法解决RAW相关
 - 通过寄存器重命名技术解决WAR和WAW相关

9.7

假设在包含前递机制的单发射 5 级静态流水线处理器的译码级添加了一个永远预测跳转的静态分支预测器，那么题 1 中的程序在这个处理器上执行需要花费多少时钟周期？

跳转指令执行了100次，当且仅当最后一次预测错误，而执行该段程序不涉及纠错过程，则周期数为：

$$2 \times 100 + 5 = 205$$

9.8

对于程序段

```
for (i = 0; i < 10; i++)
  for (j = 0; j < 10; j++)
    for (k = 0; k < 10; k++)
      {...}
```

计算分别使用一位 BHT 表和使用两位 BHT 表进行转移猜测时三重循环的转移猜测准确率，假设 BHT 表的初始值均为 0。

- for 1 bit BHT, jumping in and jumping out makes wrong, therefore:

$$1 - \frac{2 + 2 \times 10 + 2 \times 10 \times 10}{10 + 10 \times 10 + 10 \times 10 \times 10} = 0.8 = 80\%$$

- for 2 bit BHT, i has 3 times wrong, j,k has 3 wrongs for the first round and 1 wrong for each round after

$$1 - \frac{3 + 3 + 3 + 9 + 9 \times (10 + 1)}{10 + 10 \times 10 + 10 \times 10 \times 10} = 0.895 = 89.5\%$$

9.9

在一个 32 位处理器中实现一个 Cache 块大小为 64 字节、总容量为 32 KB 的数据 Cache，该数据 Cache 仅使用 32 位物理地址访问。请问，当分别采用直接映射、二路组相联和四路组相联的组织结构时，Cache 访问地址中 Tag、Index 和 Offset 三部分各自如何划分？

- 一个cache块的大小为 $64 = 2^6$,则自然有 $offset = 6$
- 总容量为 $32 \times 2^{10} = 2^{15}$,则块数为 $\frac{2^{15}}{2^6} = 2^9$
 - 对于全相联, index为9,tag为 $32 - 9 - 6 = 17$
 - 对于二路组相联, index为 $9 - 1 = 8$, tag为 $32 - 8 - 6 = 18$
 - 对于四路组相联, index为 $9 - 2 = 7$, tag为 $32 - 7 - 6 = 19$

9.10

假设程序动态执行过程中 load、store 指令占 40%。现在有两种数据 Cache 的设计方案，其中第一种方案的 Cache 容量小于第二种方案，因此采用第一种方案的 Cache 命中率为 85%，第二种方案的 Cache 命中率为 95%，但是采用第二种方案时处理器

的主频会比第一种低 10%。请问哪种设计方案性能更优?(假设 Cache 不命中情况下会阻塞流水线 100 个时钟周期)

- 记第一种方案的主频是 f , PCI为 t , 指令数为 n
- 第一种方案所需时间

$$\frac{(1 - 85\%) \times 100 \times 0.4n + nt}{f} = (6 + t)\frac{n}{f}$$

- 第二种方案所需的时间

$$\frac{(1 - 95\%) \times 100 \times 0.4n + nt}{0.9f} = \left(\frac{20 + 10t}{9}\right)\frac{n}{f}$$

故比较二者大小, 当 $t < 34$ 时, 第二种方案更优, 一般情况下很难出现一套指令集的 PCI超过34的情况, 基本上可以认为第二种设计方案更优