

“计算机组织结构” 作业 11

1. 一时钟速率为 2.5GHz 的流水式处理器执行一个有 1.5 百万条指令的程序。流水线有 5 段并以每时钟周期 1 条的速率发射指令。不考虑转移指令和无序执行所带来的性能损失。

- a) 同样执行这个程序，该处理器比非流水式处理器加速了多少？
b) 此流水式处理器的吞吐率是多少（以 MIPS 为单位）？

a) $s = (1.5 \times 10^6 \times 5) / (1.5 \times 10^6 + 5 - 1) = 5$

该处理器比非流水式处理器加速 5 倍

b) 吞吐率 = $(1.5 \times 10^6) / (1.5 \times 10^6 + 5 - 1) = 2500 \text{MIPS}$

2. 一时钟速率为 2.5GHz 的非流水式处理器，其平均 CPI 是 4。此处理器的升级版引入了 5 段流水。然而，由于锁存延迟等因素，使得处理器的时钟频率降到 2GHz。

- a) 如果不考虑转移等因素，新版处理器的加速比是多少？（相对于新版处理器不采用指令流水的情形）
b) 新、旧两版处理器的 MIPS 速率各是多少？

- a) 新版采用指令流水后，N 个指令总执行时间 $T_1 = (N+5-1)/2G = (N+4)/(2 \times 10^{-9}) \text{s}$
不用指令流水时，N 个指令总执行时间 $T_2 = (4N)/2.5G = (4N)/(2.5 \times 10^{-9}) \text{s}$
因此相对于新版处理器不采用流水的情况，加速比是：

$$T_2/T_1 = ((4N)/(2.5 \times 10^{-9})) / ((N+4)/(2 \times 10^{-9}))$$

$$= 3.2N/(N+4)$$

当 N 很大时加速比约为 3.2

- b) 新版采用流水，于是 CPI 可以等同于 1

MIPS 速率 = $2000 \text{MHz} / 1 = 2000 \text{MIPS}$

旧版不采用流水，CPI 给出为 4

MIPS 速率 = $2500 \text{MHz} / 4 = 625 \text{MIPS}$

3. 考虑一个通过指令流水线来处理的长度为 n 的指令序列。假设遇到一条有条件或无条件转移指令的概率为 p，并假设执行转移 I 时转移到非连续地址的概率是 q。请重新写出使用 k 段流水线执行 n 条指令所需总时间的公式和加速比公式。

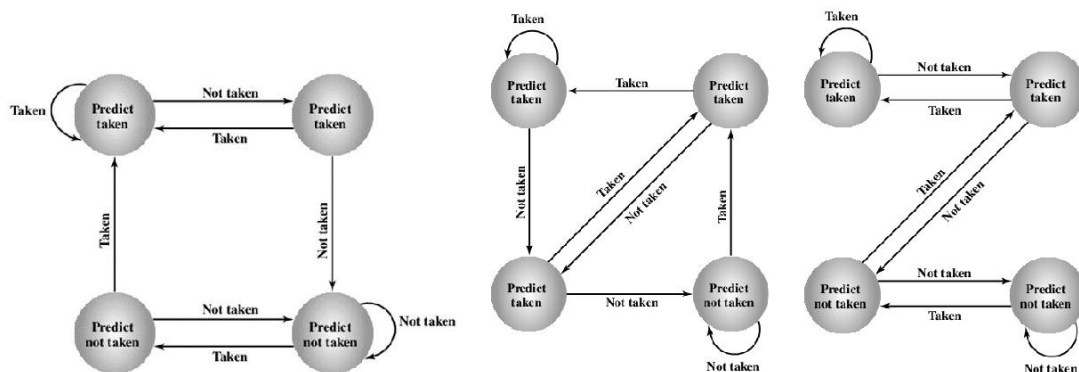
（为简化问题，认为只当发生转移的指令 I 在流水线上最后一段刚一出现时，总清流水线并撤销线上正在进行的指令。）

总时间： $(1-p) * (k+n-1) + p * (1-q) * (k+n-1) + p * q * n * k$

= $((1-p*q) * (k+n-1) + p*q*n*k) t$

加速比： $n*k / ((1-p*q) * (k+n-1) + p*q*n*k)$

4. 假设使用下面 3 种转移处理状态图



执行以下一段程序

```

int sum (int N) {
    int i, j, sum = 0;
    for (i = 0; i < N; i++)
        for (j = 0; j < N; j++)
            sum = sum + 1;
    return sum;
}

```

相应的汇编程序段为

...	
Loop-i: beq \$t1, \$a0, exit-i	# 若 (i=N) 则跳出外循环
add \$t2, \$zero, \$zero	# j=0
Loop-j: beq \$t2, \$a0, exit-j	# 若 (j=N) 则跳出内循环
addi \$t2, \$t2, 1	# j=j+1
addi \$t0, \$t0, 1	# sum=sum+1

```

        j Loop-j
exit-j: addi $t1, $t1, 1          # i=i+1
        j Loop-i
exit-i: ...

```

假设算法从流程图的左上角开始，分析 $N=10$ 和 $N=100$ 时的预测正确率。

[刘璟, 121250083]

$N=10$ 时：

一共预测了 100 次，一共有 18 次 not taken

正确率为 82%

$N=100$ 时：

一共预测了 10000 次，一共有 180 次 not taken

正确率为 98.2%

5. 有一个 ALU 不能做减法，但它能加两个输入寄存器并能对两个寄存器的各位取逻辑反。其中，数据以二进制补码形式存储。请列出用该 ALU 实现减法时控制器必须完成的操作。

被减数在 R1 中，减数在 R2 中

取反操作为 REVERSE()

1、 $MAR \leftarrow (IR(address))$

2、 $MBR \leftarrow Memory$

3、 $Y \leftarrow (MBR)$

4、 $Y \leftarrow REVERSE(Y)$

5、 $X \leftarrow 1$

6、 $Y \leftarrow (X)+(Y)$

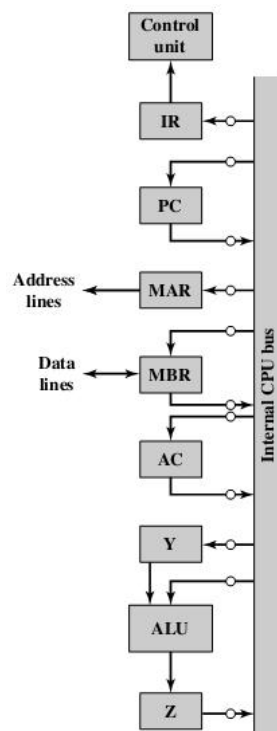
7、 $MAR \leftarrow (IR(address1))$

8、 $MBR \leftarrow Memory$

9、 $X \leftarrow (MBR)$

10、 $AC \leftarrow (X)+(Y)$

6. 如下图所示，假设沿总线和通过 ALU 的传播延迟分别为 20ns 和 100ns。由总线将数据拷贝到寄存器需要 10ns。



请问以下操作需要的最少时间为多少？

- a) 将数据从一个寄存器传送到另一个寄存器；
- b) 增量程序计数器。

- a) 两个寄存器通过总线相连
总线上传播数据需要 20ns
修改寄存器内容需要 10ns
一共 30ns
- b) 从 PC 读取地址并在总线上传输需要 20ns
修改 Y 寄存器中数据需要 10ns
放入 ALU 运算再传播需要 100ns
数据传回总线并传输需要 20ns
修改 PC 内容需要 10ns
一共用时 160ns

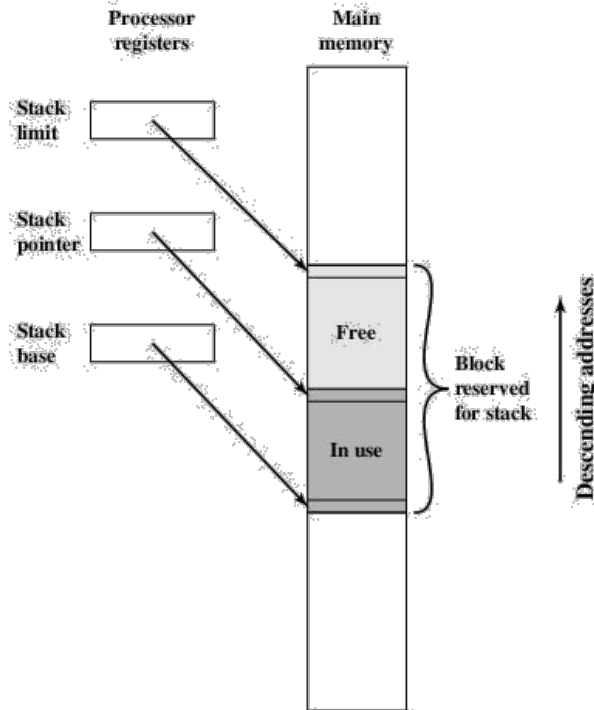
7. 以题 6 中的图为例，加一个数到 AC。请写出该数为以下情形分别所需要的微操作序列：

- a) 立即数
- b) 直接寻址的操作数
- c) 间接寻址的操作数

[袁睿, 131250088]

- a) 1、 $Y \leftarrow (IR(address))$
2、 $Z \leftarrow (AC) + (Y)$
3、 $AC \leftarrow (Z)$
- b) 1、 $MAR \leftarrow (IR(address))$
2、 $MBR \leftarrow Memory$
3、 $Y \leftarrow MBR$
4、 $Z \leftarrow (AC) + (Y)$
5、 $AC \leftarrow (Z)$
- c) 1、 $MAR \leftarrow (IR(address))$
2、 $MBR \leftarrow Memory$
3、 $MAR \leftarrow (MBR)$
4、 $MBR \leftarrow Memory$
5、 $Y \leftarrow (MBR)$
6、 $Z \leftarrow (AC) + (Y)$
7、 $AC \leftarrow (Z)$

8. 下图所示的栈保存在内存中，寄存器中存储了栈限（分配给该栈的最小地址）、栈指针（栈顶地址）和栈基（分配给该栈的最大地址）。请写出 push 和 pop 该栈所对应的微操作序列。



POP: $SP \leftarrow (SP)+1$

PUSH: 1、 $SP \leftarrow (SP)-1$

2、 $MBR \leftarrow (R1)$ //需要入栈的数据放在 R1 寄存器中

3、 $MAR \leftarrow (SP)$

4、 $Memory \leftarrow (MBR)$

9. 一个指令周期有 4 个主要阶段：取指、间址、执行和中断。硬布线方式实现时，采用一个 2 位的寄存器来标志当前阶段，但微程序式控制器却不需要类似的标志。请问为什么硬布线式控制器需要这些标志，而微程序式控制器不需要这些标志？

硬布线式控制器在运行时，处理当前指令执行到了哪一阶段的信息，会作为输入的一部分，用于布尔逻辑计算，明确标记当前执行的阶段

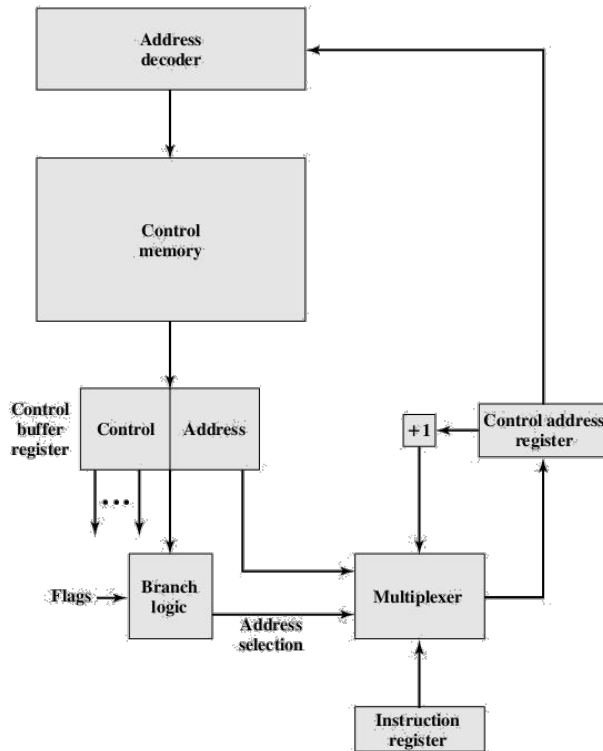
所有的微操作都在微程序式控制器中，相关逻辑会把一系列微操作见的顺序确定下来，因此不需要状态标志

10. 控制器如下图所示。假定它的控制存储器是 24 位宽。微指令格式的控制部分分成两个字段。一个 13 位的微操作字段用来指定将要完成的微操作。一个地址选择字段用来指明能引起微指令转移的条件，这些条件是基于 8 个标志来建立的。

- 地址选择字段有多少位？
- 地址字段有多少位？
- 控制存储器容量为多少？
- 无条件转移指令应该如何完成？
- 如何避免转移，即描述一条不指定任何（有条件的和无条件的）转移的微指令？

- 地址标志一共 8 个，需要 3 位，所以地址选择字段有 3 位
- 地址字段位数=总宽-微操作字段位数-地址标志位数=24-13-3=8 位
- 地址字段位数有 8 位，表示微指令最多 $2^8=256$ 条，所以控制存储器容量为 $256 \times 24/8=768B$
- 可以 8 个标志里取一个作为无条件转移的标志，或者规定某个特定地址为无条件转移地址

e) 规定一个特定地址选择字段，表示不发生转移



11. CPU 有 16 个寄存器，一个 ALU 有 16 种逻辑功能和 16 种算术功能，一个移位器有 8 种操作，所有这些组件都与一个 CPU 内部总线相连。设计一种微指令格式能指定此 CPU 的各种微操作。

逻辑功能需要 4 位，算术功能需要 4 位，可以放在一起一共 5 位

移位器需要 3 位

假定输入输出的数据都位于 16 个寄存器里，则输入和输出都需要 4 位

于是微指令格式如下：

逻辑功能和算术功能（0-4）+ 移位操作（5-7）+ ALU 输入 1（8-11）+ ALU 输入（12-15）+ ALU 输出（16-19）

12. 使用一种编码式微指令格式。说明如何划分 9 位的微操作字段，恰好能指定 46 种不同动作。

9 位操作一共可以指定 $2^9=512$ 条指令

可以对指令中的位区域进行划分，分成两部分

如左边 4 位，右边 5 位

当左边指定了 $2^4-1=15$ 种操作时，右边全 0

当右边指定了 $2^5-1=31$ 种操作时，左边全 0

一共有 46 种不同动作，符合题意