

## “计算机组织结构”作业 05 参考答案

1. 假定某计算机中有一条转移指令，采用相对寻址方式，共占 2 个字节，第一字节是操作码，第二字节是相对位移量（用补码表示），CPU 每次从内存只能取一个字节。假设执行到某转移指令时 PC 的内容为 200，执行该转移指令后要求转移到 100 开始的一段程序执行，则该转移指令第二字节的内容应该是多少（二进制表示，需要在末尾加 B）？

$100=200+2+Offset$ ,  $Offset=100-202=-102=10011010B$

2. 假设地址为 1200H 的内存单元中的内容为 120CH，地址为 120CH 的内存单元的内容为 38B8H，而 38B8H 单元的内容为 88F9H。说明以下各情况下操作数的操作数是多少（十六进制表示，需要在末尾加 H）？[陈姿丽，121250018]

- a) 操作数采用变址寻址，变址寄存器的内容为 12，指令中给出的形式地址为 1200H。  
b) 操作数采用一次间接寻址，指令中给出的地址码为 1200H。  
c) 操作数采用寄存器间接寻址，指令中给出的寄存器编号为 8，8 号寄存器的内容为 1200H。

a) 有效地址  $EA=000CH+1200H=120CH$  操作数为 38B8H

b) 有效地址  $EA=(1200H)=120CH$  操作数为 38B8H

c) 有效地址  $EA=1200H$  操作数为 120CH

3. 考虑一个 16 位处理器，它的一条装入指令以如下情况出现在主存，起始地址为 200。

200	Load to AC	Mode
201	500	
202	下一条指令	

第一字的第一部分指出此指令是将一个值装入累加器。Mode 字段用于指定一种寻址方式。若寻址方式需要的话，Mode 字段拨出一部分指定源寄存器；这里假定使用的源寄存器是 R1，有值 400。还有一个基址寄存器，它有值 100。地址 201 处的值 500，可以是立即数也可以是地址计算的一部分。假定位置 399 处有值 999，位置 400 处有值 1000，如此等等。请对如下寻址方式确定将被装入的操作数（十进制表示）：

- a) 直接      b) 立即      c) 间接      d) PC 相对      e) 基址  
f) 寄存器      g) 寄存器间接      h) 变址（用 R1 自动增量）

[潘琦，121250105]

a) 有效地址  $EA=500$  被装入的操作数为 1100

b) 有效地址  $EA=201$  被装入的操作数为 500

c) 有效地址  $EA=(500)=1100$  被装入的操作数为 1700

d) 有效地址  $EA=200+2+500=702$  被装入的操作数为 1302

e) 有效地址  $EA=100+500=600$  被装入的操作数为 1200

f) 有效地址  $EA=R1$  被装入的操作数为 400

g) 有效地址  $EA=(R1)=400$  被装入的操作数为 1000

h) 有效地址  $EA=500+400=900$  被装入的操作数为 1500

4. 若 CPU 取并执行一条间接地址方式指令，指令是：(a) 一个要求单操作数的计算；(b) 一个转移，CPU 需要访问存储几次？

a) CPU 访问主存 3 次

1. CPU 取指令访问主存 1 次；2. CPU 间接寻址取得操作数需访问主存 2 次（因为是单操作数，所以是  $AC \leftarrow AC + Y$ ，所以无存回操作）

[王子安, 141250146]

b) CPU 访问主存 2 次

1. CPU 取指令访问主存 1 次；2. CPU 取转移地址访问主存 1 次

5. 考虑一个包括基址带变址寻址方式的处理器。假设遇到使用这种寻址方式的一条指令，指令给定的偏移量是 1970（十进制）。当前的基址和变址寄存器分别有十进制数 48022 和 8。操作数的地址是什么（十进制表示）？

操作数地址为  $48022 + 8 + 1970 = 50000$

6. 一 PC 相对寻址方式的转移指令存于地址为 620（十进制）的存储器位置中。它要转移到 530（十进制）位置上。指令长度为 32 位，其中地址字段长度是 10 位，其二进制值是什么（二进制表示，需要在末尾加 B）？

执行到转移指令时，PC 已经完成自增，值为  $620 + 4 = 624$ ，则  $offset = 530 - 624 = -94$ 。

由于指令中地址段长度为 10 位，二进制表示为 **1110100010B**。

7. 设计一种变长操作码，以允许如下全都能编码成 36 位指令中：

指令有两个 15 位地址和一个 3 位寄存器号

指令有一个 15 位地址和一个 3 位寄存器号

指令没有地址或寄存器

指令字 (3)	地址 1 (15)	地址 2 (15)	寄存器 (3)
---------	-----------	-----------	---------

其中指令字可使用 010, 011, 100, 101, 110, 111

指令字 (18)	地址 2 (15)	寄存器 (3)
----------	-----------	---------

其中指令字以 001 开头

指令字 (36)
----------

其中指令字以 000 开头

8. 定义：

$EA = (X) +$  是有效地址等于位置 X 的内容，并在有效地址计算后 X 增加 1 字长；

$EA = -(X)$  是有效地址等于位置 X 的内容，并在有效地址计算前 X 减少 1 字长；

$EA = (X) -$  是有效地址等于位置 X 的内容，并在有效地址计算后 X 减少 1 字长。

考虑如下指令，它们都有（操作，源操作数，目的操作数）的格式，并操作结果放入目的操作数。

- a) OP X, (X)      b) OP (X), (X) +      c) OP (X) +, (X)
- d) OP - (X), (X)      e) OP - (X), (X) +      f) OP (X) +, (X) +
- g) OP (X) -, (X)

使用 X 作为堆栈指针，上述哪些指令能由堆栈弹出顶部两元素，完成所要求的操作（例如，ADD 源到目的并存入目的），并将结果压回堆栈？（按序列举出所有符合要求的指令编号，不需要括号或任意形式的间隔）

**cg**

a) 源操作数是 X 中的内容，目的操作数是 (X) 中的内容，不符合

b) 源操作数和目的操作数都是 (X) 中的内容，不符合

c) 源操作数是 (X) 中的内容，目的操作数是 (X+1) 中的内容，结果会存在 (X+1) 处，符合

(此时栈是朝 0 地址方向增长)

- d) 源操作数和目的操作数都是 (X-1) 中的内容, 不符合
- e) 源操作数和目的操作数都是 (X-1) 中的内容, 不符合
- f) 源操作数是 (X) 中的内容, 目的操作数是 (X+1) 中的内容, 但结果会存在 (X+2) 处, 不符合
- g) 源操作数是 (X) 中的内容, 目的操作数是 (X-1) 中的内容, 结果会存在 (X-1) 处, 符合  
(此时栈是朝高地址方向增长)

9. 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。

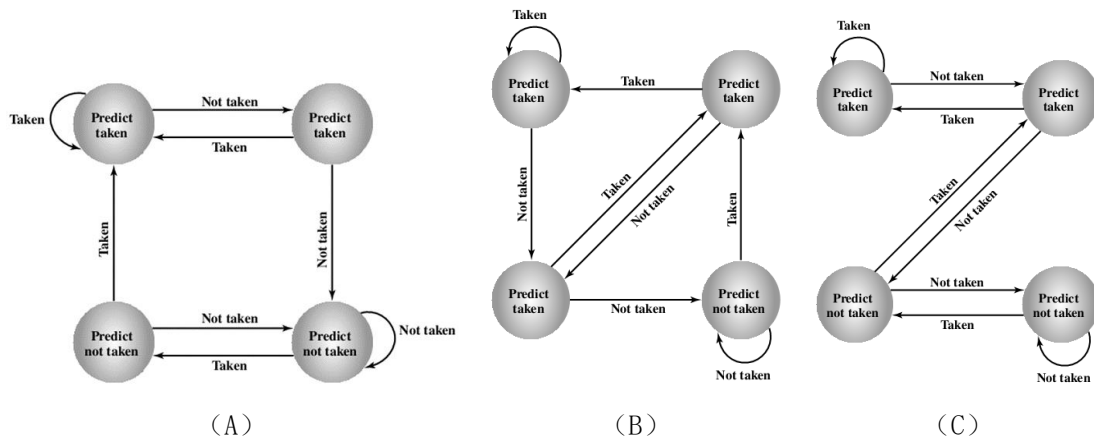
- a) 同样执行这个程序, 该处理器比非流水式处理器加速了多少 (百分数)?
- b) 此流水式处理器的吞吐率是多少 (以 MIPS 为单位)?

a) 加速比  $S_k = \frac{nkt}{[k+(n-1)]t} = \frac{k}{1+\frac{k-1}{n}}$

由于有 1.5 百万条指令, 即 n 很大, 所以  $S_k$  为 k, 即 5, 加速了 400%。

b) 由于近似于每个周期完成一条指令, 所以吞吐率为  $2.5G/10^6=2500$  MIPS。

10. 假设使用下面 3 种转移处理状态图 A、B、C



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  $N=10$  时，使用转移处理状态图 A 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=10$  时，使用转移处理状态图 A 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 A 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 A 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=10$  时，使用转移处理状态图 B 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=10$  时，使用转移处理状态图 B 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 B 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 B 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=10$  时，使用转移处理状态图 C 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=10$  时，使用转移处理状态图 C 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 C 的外层 for 循环预测正确率（百分数，精度：小数点后 2 位）。
- 分析  $N=100$  时，使用转移处理状态图 C 的内层 for 循环预测正确率（百分数，精度：小数点后 2 位）。

外循环共预测  $N+1$  次，内循环共预测  $N \times (N+1)$  次。外循环和内循环各有一组预测位。

使用转移处理状态图 A 时：

预测初始位为 11，外循环中第 1 次、第 2 次和最后一次预测错误，共错误 3 次。内循环中第 1 次进入时变成 10（不发生，预测错误），然后变成 00（不发生，预测错误），跳出时又变成 10（发生，预测错误）；其后每次进入时变成 00（不发生，预测正确），跳出时又变成 01（发生，预测错误），所以内循环共有  $N+2$  次预测错误。

- $N=10$ ：外循环正确率  $1-3/11=72.73\%$
- $N=10$ ：内循环正确率  $1-12/110=89.09\%$
- $N=100$ ：外循环正确率  $1-3/101=97.03\%$
- $N=100$ ：内循环正确率  $1-102/10100=98.99\%$

使用转移处理状态图 B 时：

预测初始位为 11，外循环中第 1 次、第 2 次和最后一次预测错误，共错误 3 次。内循环中第 1 次进入时变成 01（不发生，预测错误），然后变成 00（不发生，预测错误），跳出时又变成 10（发生，预测错误）；其后每次进入时变成 01（不发生，预测错误），然后变成 00（不发生，预测错误），跳出时又变成 10（发生，预测错误），所以内循环共有  $3N$  次预测错误。

- e)  $N=10$ : 外循环正确率  $1-3/11=72.73\%$
- f)  $N=10$ : 内循环正确率  $1-30/110=72.73\%$
- g)  $N=100$ : 外循环正确率  $1-3/101=97.03\%$
- h)  $N=10$ : 内循环正确率  $1-300/10100=97.03\%$

使用转移处理状态图 C 时：

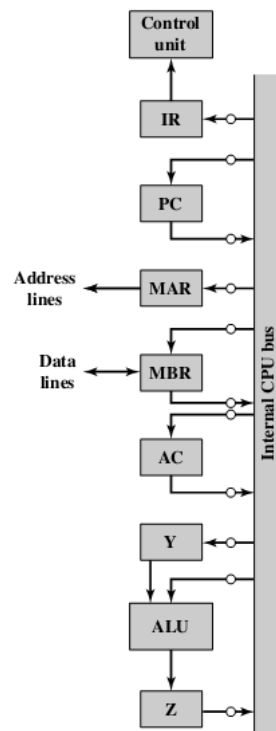
预测初始位为 11，外循环中第 1 次、第 2 次和最后一次预测错误，共错误 3 次。内循环中第 1 次进入时变成 10（不发生，预测错误），然后变成 01（不发生，预测错误），然后变成 10（不发生，预测正确），跳出时又变成 01（发生，预测错误）；其后每次进入时变成 00（不发生，预测正确），跳出时又变成 01（发生，预测错误），所以内循环共有  $N+2$  次预测错误。

- i)  $N=10$ : 外循环正确率  $1-3/11=72.73\%$
- j)  $N=10$ : 内循环正确率  $1-12/110=89.09\%$
- k)  $N=100$ : 外循环正确率  $1-3/101=97.03\%$
- l)  $N=100$ : 内循环正确率  $1-102/10100=98.99\%$

[伍佳艺, 141250150]

注：中间那张处理状态图的上面两个状态实际上可以合并[朱宇翔, 141250216]

11. 如下图所示，假设沿总线 and 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；

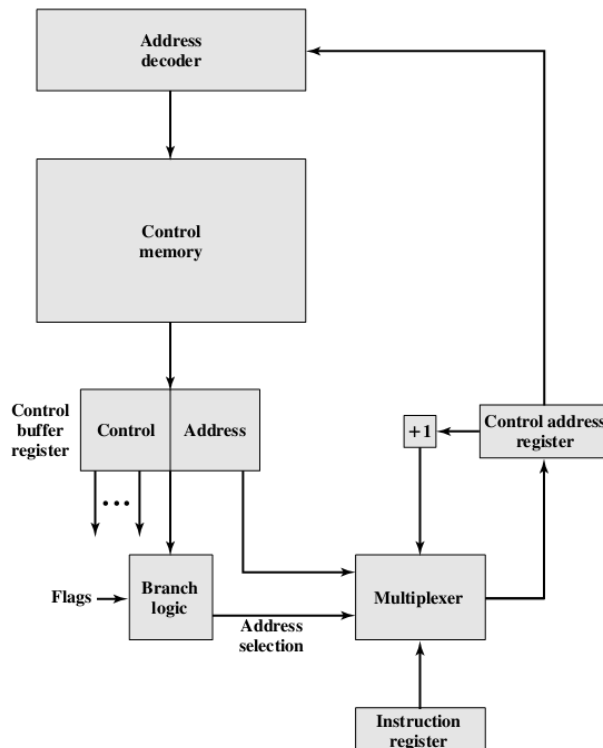
在总线上传送数据用时 20ns，拷贝到寄存器中需要 10ns，总共需要 30ns

b) 使用 ALU 增量程序计数器。

从 PC 中读取地址传送到总线用时 20ns, 放入 Y 中用时 10ns; 经 ALU 中计算用时 100ns; 将数据传输回 PC 用时 20ns+10=30ns; 所以总共用时 160ns。

12. 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。

- a) 地址选择字段有多少位?
- b) 地址字段有多少位?
- c) 控制存储器容量为多少 (单位: 字节)?



- a) 地址选择字段共有 8 个标志,  $\log_2 8 = 3$ , 故需要 3 位
- b) 地址字段位数为  $24 - 13 - 3 = 8$  位
- c) 地址为 8 位, 表示微指令的条数最大为  $2^8 = 256$  条, 所以控制存储器的容量为  $256 * 24 / 8 = 768$  字节

13. 典型的微处理器是用不同地址去访问指定设备控制器中的 I/O 数据寄存器、控制和状态寄存器。这些寄存器被称为端口 (port)。Intel 8088 使用两类 I/O 指令格式。一类格式中, 8 位操作码指定 I/O 操作, 后随 8 为端口地址; 另一类格式中, I/O 操作码隐含指示端口地址在 16 位的 DX 寄存器中。

- a) 第一类寻址模式下, 8088 能寻址多少端口?
- b) 第二类寻址模式下, 8088 能寻址多少端口?

采用直接寻址方式的端口地址为 1 个字节长, 可寻址  $2^8$  即 256 个端口

采用间接寻址方式的端口地址为两个字节长, 可寻址  $2^{16}$  即 65536 个端口

14. 在程式 I/O 中，处理器陷入一个等待循环来检查 I/O 设备状态。为提高效率，可编写这样的 I/O 软件：处理器周期性地检查设备状态，若设备未就绪，处理器跳转到执行其他任务，在某指定时间长度的间隔后处理器再次检查设备状态。
- a) 考虑采用上述方法向打印机一次 1 字符地输出数据。打印机以 10 字符/秒速度运行。若每 20ms 扫描一次它的状态，需要查询几次才能得到获得打印机就绪状态？
  - b) 接着考虑一个具有单一字符缓冲器的键盘。平均而言，字符以 10 字符/秒的速度被敲入。然而两次连续按键的时间间隔是 60ms。I/O 程序应该以什么频率扫描键盘？（单位：次/秒）
- a) 由于打印机的输出速度为 10 字符/秒，输出一个字符所需要的时间为  $1/10=0.1s=100ms$ 。若处理器每隔 20ms 扫描一次打印机状态，则需要查询 5 次才能得到获得打印机就绪状态。
- b) 虽然字符以 10 字符/秒的速度敲入，但两次连续按键的时间间隔为 60ms，且键盘只有单一字符的缓冲器。为了保证之前输入的字符不丢失，I/O 程序扫描键盘的间隔应该不大于 60ms，即频率至少为 17 次/秒。
15. 考虑某系统对一个设备使用了中断驱动式 I/O，此设备以平均 64kbps 的速度连续传送数据。
- a) 假设中断处理大约用  $100\mu s$ ，每字节中断一次，确定处理器时间的百分之几被这个 I/O 设备所消耗。（百分数，精度：整数）
  - b) 假设这个设备有两个 16 字节的缓冲器并当一个缓冲器满时才中断处理器一次。执行此中断服务时，处理时间需要延长（每传送一个字节处理器要用大约  $8\mu s$ ），重复上问。（百分数，精度：整数）
- a) 由于每个字节中断一次，而数据传输率为 64kbps，所以每秒钟的中断次数为 8k 次。因此，该 I/O 设备占用处理器时间的比例是  $8k \times 100\mu s / 1s = 80\%$
- b) 处理一个缓冲器的中断耗时  $100 + (16-1) \times 8 = 220\mu s$ ，而每秒钟处理的次数为  $8kB / 16B = 500$  次。因此，该 I/O 设备占用处理器时间的比例是  $220 \times 500\mu s / 1s = 11\%$
16. 在一个系统中，经由总线的一次数据传送用 500ns。总线控制的传递，无论是由处理器到 DMA 模块还是由 DMA 模块到处理器，两个方向上都是用 200ns。一个有 400kbps 数据传输率的 I/O 设备使用了 DMA 来传送 128 字节的块，每次传送 1 字节的数据。
- a) 若使用突发模式 DMA，即块传送之前 DMA 模块获得总线控制权并一直维持对总线的控制直到整个块传送完。设备占用总线多长时间（含获取总线控制和交回总线控制）？（单位：毫秒，精度：小数点后 4 位）
  - b) 若使用周期窃取式 DMA，重复上问。
- a) DMA 获取总线控制和交回总线控制的时间均为 200ns。在总线上传输前 127 字节需要的时间为  $500ns \times 127B / 1B = 63500ns$ ，而 I/O 设备提供数据的时间为  $128B / 400kbps = 2.56ms$ 。因此，总线占用时间为：  
 $200ns + 0.02ms + \max(2.54ms, 63500ns) + 500ns + 200ns = 2.5609ms$   
[晁贝贝, 121250007][梁家诚, 141250068]
- b) 使用周期窃取式 DMA 时，DMA 每次进行数据传输都要获取和交回总线，所以设备占用总线的时间为： $128B / 1B \times (200 + 500 + 200) = 0.1152ms$
17. 假设一个主频为 1GHz 的处理器需要从某个成块传送的 I/O 设备读取 1000 字节的数据到主存缓冲区中，该 I/O 设备一旦启动即按 400kbps 的数据传输率向主机传送 1000 字节



数据，每个字节的读取、处理并存入内存缓冲区需要 1000 个时钟周期，则以下 4 种方式下，在 1000 字节的读取过程中，

- 采用查询方式，每次处理一个字节，一次状态查询至少需要 60 个时钟周期。CPU 用在该设备的 I/O 操作上的时间为多少？（单位：毫秒，精度：小数点后 3 位）
- 问题 a) 中 CPU 处理 I/O 时间占整个处理器时间的百分比为多少？（百分数，精度：小数点后 2 位）
- 采用中断驱动 I/O 方式，外设每准备好一个字节发送一次中断请求。每次中断响应需要两个时钟周期，中断服务程序的执行需要 1200 个时钟周期。CPU 用在该设备的 I/O 操作上的时间为多少？（单位：毫秒，精度：小数点后 3 位）
- 问题 c) 中 CPU 处理 I/O 时间占整个处理器时间的百分比为多少？（百分数，精度：小数点后 2 位）
- 采用周期挪用 DMA 方式，每挪用一次主存周期处理一个字节，DMA 初始化和后处理的时间为 2000 个时钟周期，CPU 和 DMA 没有访存冲突。CPU 用在该设备的 I/O 操作上的时间为多少？（单位：毫秒，精度：整数）
- 问题 e) 中 CPU 处理 I/O 时间占整个处理器时间的百分比为多少？（百分数，精度：小数点后 2 位）

- I/O 设备的数据传输率为 400kbps，因此传输每个字节的时间为  $1\text{B}/400\text{kbps}=20\mu\text{s}$ 。处理器主频为 1GHz，所以每个时钟周期为 1ns。考虑最坏的情形，数据到达时刚刚查询过，需要再等 60 个时钟周期才会查询，加上每个字节读取、处理和存入内存缓冲的时间为  $(60+1000)\times 1\text{ns}=1.06\mu\text{s}<20\mu\text{s}$ 。因此 CPU 完成整个数据传输的时间为  $1000\text{B}/400\text{kbps}+1\mu\text{s}=20.001\text{ms}$ 。

[刘丞, 121250078][袁睿, 131250088]

- 在查询方式中，CPU 会一直参与，因此占用处理器时间的百分比为 100.00%。
- 采用中断驱动方式，占用 CPU 的总时间为  $(2+1200)\times 1\text{ns}\times 1000=1.202\text{ms}$ 。
- I/O 操作占整个处理器时间的百分比为：  
 $1.202\text{ms}/(1000\text{B}/400\text{kbps}+(2+1200)\times 1\text{ns})=6.01\%$ 。 [曹雨婷, 121250005]  
 （注意：此处中断处理包含了数据传输。）
- 在周期挪用方式下，由于 CPU 和 DMA 没有访存冲突，则 CPU 受到影响的时间为  $2000\times 1\text{ns}\times 1000=2\text{ms}$ 。
- I/O 操作上占整个处理器时间的百分比为：  
 $2\text{ms}/(1000\text{B}/400\text{kbps}+(2000+1000)\times 1\text{ns})=10.00\%$

[卢苇, 121250091][马昕, 131250093][刘子龙, 181830128][王崇羽, 181830175]

===== 分割线：以下内容不在小程序上提交 =====

18. 以 0-、1-、2-、3-地址法分别编写程序来计算：

$$X = (A + B \times C) / (D - E \times F)$$

0 地址	1 地址	2 地址	3 地址
PUSH M	LOAD M	MOV (X<-Y)	MOVE (X<-Y)
POP M	STORE M	ADD (X<-X+Y)	ADD (X<-Y+Z)
ADD	ADD M	SUB (X<-X-Y)	SUB (X<-Y-Z)
SUB	SUB M	MUL (X<-X×Y)	MUL (X<-Y×Z)
MUL	MUL M	DIV (X<-X/Y)	DIV (X<-Y/Z)



DIV	DIV M		
-----	-------	--	--

其中，0 地址法是采用了堆栈，每次对堆栈顶端的两个数进行操作，例如 ADD 实际上是用堆栈次顶端的数加上堆栈顶端的数。

0 地址	1 地址	2 地址	3 地址
PUSH A	LOAD E	MOV R0, E	MUL R0, B, C
PUSH B	MUL F	MUL R0, F	ADD R0, A, R0
PUSH C	STORE P	MOV R1, D	MUL R1, E, F
MUL	LOAD D	SUB R1, R0	SUB R1, D, R1
ADD	SUB P	MOV R0, B	DIV X, R0, R1
PUSH D	STORE P	MUL R0, C	
PUSH E	LOAD B	ADD R0, A	
PUSH F	MUL C	DIV R0, R1	
MUL	ADD A	MOV X, R0	
SUB	DIV P		
DIV	STORE X		
POP X			

19. 某计算机指令系统采用定长指令字格式，指令字长 16 位，每个操作数的地址码长 6 位。指令分为 2 地址、1 地址和 0 地址三类。如果 2 地址的指令有  $k_2$  条，0 地址的指令有  $k_0$  条，那么 1 地址的指令最多有多少条？（提示：任何指令不能有二义性，即任何指令无法同时用 2-、1-、0-地址法中两种或两种以上方式解释。）[刘璟, 121250083]

为了避免指令的二义性，要求同一条指令不能同时可能被解释为 2 地址、1 地址和 0 地址中的两种或三种。对于 1 地址指令，操作数长度为 6 位，因此操作码长度为 10 位。由于 2 地址指令共有  $k_2$  种，所以前 10 位的取值中有  $k_2 \times 2^6$  种可以被解释为 2 地址指令；由于 0 地址指令共有  $k_0$  种，所以前 10 位的取值中至少有  $\lceil k_0/2^6 \rceil$  种可以被解释为 0 地址指令。所以 1 地址指令最多有  $k_1 = 2^{10} - k_2 \times 2^6 - \lceil k_0/2^6 \rceil$ 。

补充：

$$k_0 = 2^{16} - k_2 \times 2^{12} - k_1 \times 2^6。$$

$$k_2 = 2^4 - \left\lceil \left( \frac{k_0}{2^6} + k_1 \right) / 2^6 \right\rceil。$$

[熊禧华, 191250161]

20. 假设某个计算机只有一条指令：

SUBS X      累加器减去位置 X 处的内容，结果存入累加器和位置 X 处。

通过编程实现以下功能：

- 将位置 X 处的数据传输到累加器
- 将累加器的数据传输到位置 X 处
- 将位置 X 处的内容加到累加器

（提示：可以使用 1 个或多个内容为 0 的位置 Y、Z……）

假设 AC 处的初始值为 a，X 处的值为 x，Y、Z 处的初始值为 0

- a) SUBS Y; // AC = a, X = x, Y = a  
 SUBS Y; // AC = 0, X = x, Y = 0  
 SUBS X; // AC = -x, X = -x, Y = 0  
 SUBS Y; // AC = -x, X = -x, Y = -x  
 SUBS Y; // AC = 0, X = -x, Y = 0  
 SUBS X; // AC = x, X = x, Y = 0
- b) SUBS Y; // AC = a, X = x, Y = a  
 SUBS X; // AC = a-x, X = a-x, Y = a  
 SUBS X; // AC = 0, X = 0, Y = a  
 SUBS Y; // AC = -a, X = 0, Y = -a  
 SUBS X; // AC = -a, X = -a, Y = -a  
 SUBS Y; // AC = 0, X = -a, Y = 0  
 SUBS X; // AC = a, X = a, Y = 0
- c) SUBS Y; // AC = a, X = x, Y = a, Z = 0  
 SUBS Z; // AC = a, X = x, Y = a, Z = a  
 SUBS Y; // AC = 0, X = x, Y = 0, Z = a  
 SUBS X; // AC = -x, X = -x, Y = 0, Z = a  
 SUBS Z; // AC = -x-a, X = -x, Y = 0, Z = -x-a  
 SUBS Y; // AC = -x-a, X = -x, Y = -x-a, Z = -x-a  
 SUBS Y; // AC = 0, X = -x, Y = 0, Z = -x-a  
 SUBS Z; // AC = x+a, X = -x, Y = 0, Z = x+a

21. 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。）

	Time →							← Branch penalty						
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Instruction 1	FI	DI	CO	FO	EI	WO								
Instruction 2		FI	DI	CO	FO	EI	WO							
Instruction 3			FI	DI	CO	FO	EI	WO						
Instruction 4				FI	DI	CO	FO							
Instruction 5					FI	DI	CO							
Instruction 6						FI	DI							
Instruction 7							FI							
Instruction 15								FI	DI	CO	FO	EI	WO	
Instruction 16									FI	DI	CO	FO	EI	WO

$$\text{总时间 } T = [k + (n - 1)]t + pqn(k - 2)t$$

$$\text{加速比 } S_k = \frac{nkt}{[k + (n - 1)]t + pqn(k - 2)t} = \frac{nk}{k + n - 1 + pqn(k - 2)}$$

22. 根据题 12，回答以下问题：

- a) 无条件转移指令应该如何完成?
- b) 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?
- a) 如果在 8 位标志中, 有 1 位用于标志无条件转移, 则该位为 1 时进行无条件转移; 如果没有相应的标志, 可以规定某个特定的地址选择字段(如 111)表示无条件转移, 此时地址字段为转移地址。
- b) 可以规定某个特定地址选择字段(如 000)表示不转移。

23. 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。请根据以下 4 种情形, 列出用该 ALU 实现减法时控制器必须完成的操作。

- a) 1 地址直接寻址
- b) 1 地址间接寻址
- c) 2 地址直接寻址
- d) 2 地址间接寻址

假设 ALU 的两个输入寄存器为 X 和 Y。取反操作为 Neg(), 加法为 Add()

a) 1 地址, 直接寻址

```
t1: MAR ← (IR(address))
t2: MBR ← Memory
t3: Y ← (MBR)
t4: Y ← Neg(Y)
t5: X ← 1
t6: Y ← (X) + (Y)
t7: X ← (AC)
t8: AC ← (X) + (Y)
```

b) 1 地址, 间接寻址

```
t1: MAR ← (IR(address))
t2: MBR ← Memory
t3: MAR ← (MBR)
t4: MBR ← Memory
t5: Y ← (MBR)
t6: Y ← Neg(Y)
t7: X ← 1
t8: Y ← (X) + (Y)
t9: X ← (AC)
t10: AC ← (X) + (Y)
```

c) 2 地址, 直接寻址

```
t1: MAR ← (IR(address2))
t2: MBR ← Memory
t3: Y ← (MBR)
t4: Y ← Neg(Y)
t5: X ← 1
t6: Y ← (X) + (Y)
```

```
t7: MAR  $\leftarrow$  (IR(address1))
t8: MBR  $\leftarrow$  Memory
t9: X  $\leftarrow$  (MBR)
t10: AC  $\leftarrow$  (X) + (Y)
```

d) 2 地址, 间接寻址

```
t1: MAR  $\leftarrow$  (IR(address2))
t2: MBR  $\leftarrow$  Memory
t3: MAR  $\leftarrow$  (MBR)
t4: MBR  $\leftarrow$  Memory
t5: Y  $\leftarrow$  (MBR)
t6: Y  $\leftarrow$  Neg(Y)
t7: X  $\leftarrow$  1
t8: Y  $\leftarrow$  (X) + (Y)
t9: MAR  $\leftarrow$  (IR(address1))
t10: MBR  $\leftarrow$  Memory
t11: MAR  $\leftarrow$  (MBR)
t12: MBR  $\leftarrow$  Memory
t13: X  $\leftarrow$  (MBR)
t14: AC  $\leftarrow$  (X) + (Y)
```

24. 以题 11 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

```
t1: Y  $\leftarrow$  (IR(address))
t2: Z  $\leftarrow$  (AC) + (Y)
t3: AC  $\leftarrow$  (Z)
```

b) 直接寻址的操作数

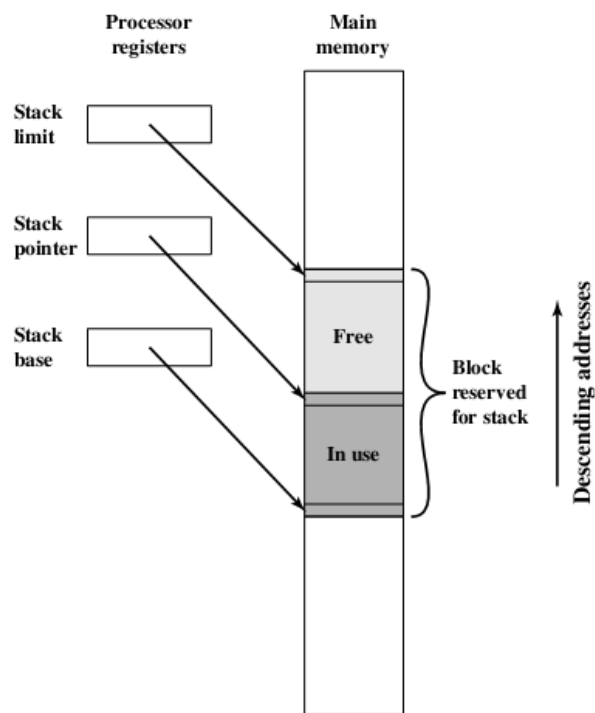
```
t1: MAR  $\leftarrow$  (IR(address))
t2: MBR  $\leftarrow$  Memory
t3: Y  $\leftarrow$  (MBR)
t4: Z  $\leftarrow$  (AC) + (Y)
t5: AC  $\leftarrow$  (Z)
```

c) 间接寻址的操作数

```
t1: MAR  $\leftarrow$  (IR(address))
t2: MBR  $\leftarrow$  Memory
t3: MAR  $\leftarrow$  (MBR)
t4: MBR  $\leftarrow$  Memory
t5: Y  $\leftarrow$  (MBR)
t6: Z  $\leftarrow$  (AC) + (Y)
t7: AC  $\leftarrow$  (Z)
```

25. 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:  t1: MAR <- (SP)
      t2: MBR <- Memory
      SP <- (SP) + 1
PUSH: t1: SP <- (SP) - 1
      t2: MAR <- (SP)
      t3: Memory <- (MBR)
```

[明鑫, 171250553]

26. 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？

在硬连线式控制器中，当前阶段会作为输入的一部分，用于布尔逻辑式的计算，因此需要采用 2 位的寄存器来标志当前阶段。

在微程序式控制器中，所有的微指令都存储在控制存储器中，排序逻辑会确定下一条将要执行的微指令，各个阶段之间通过跳转来实现，因此不需要状态标志。

27. 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。假设 ALU 的输入和输出都位于寄存器中，设计一种微指令格式能指定此 CPU 的各种微操作。

微指令格式如下：

逻辑和算术功能 (0~4位)	移位操作 (5~7位)	ALU 输入1 (8~11位)	ALU 输入2 (12~15) 位	ALU 输出 (16~19位)
-------------------	----------------	--------------------	----------------------	--------------------

28. 在包含 DMA 模块的系统中，一般 DMA 存取主存储器的优先级比处理器存取主存储器的优先级高，为什么？

如果处理器被停止访问主存储器，除处理时间延长外不产生其它问题。但 DMA 可能连接高速数据传输设备，如果不及时获取主存储器的存取权限来及时写入数据，则可能会因为缓冲区溢出而造成数据丢失。所以，一般 DMA 存取主存储器的优先级会高于处理器。

29. 假定某计算机的 CPU 主频为 500MHz，所连接的某个外设的最大数据传输率为 160kbps，该外设接口中有一个 16 位的数据缓存器，相应的中断服务程序的执行时间为 500 个时钟周期，则是否可以用中断方式进行该外设的输入输出？假定该外设的最大数据传输率改为 16Mbps，则是否可以用中断方式进行该外设的输入输出？

- a) 外设的最大传输率为 160kbps，而每传输完 16 位进行一次中断处理，因此 1 秒内的中断次数为  $160\text{kbps}/16\text{bit}=10\text{k}$  次。所以，1 秒内需要的中断处理的时间为  $10\text{k}/500\text{MHz}\times 500=10\text{ms}<1\text{s}$ ，所以可以用中断方式进行该外设的输入输出。
- b) 外设的最大传输率为 16Mbps，而每传输完 16 位进行一次中断处理，因此 1s 内的中断次数为  $16\text{Mbps}/16\text{bit}=10^6$  次。所以，1 秒内需要的中断处理的时间为  $10^6/500\text{MHz}\times 500=1\text{s}$ ，所以可以用中断方式进行该外设的输入输出。

[刘丞，121250078]

30. 若某计算机有 5 级中断，中断响应优先级为  $1>2>3>4>5$ ，而中断处理优先级为  $1>4>5>2>3$ 。要求完成以下工作：

- a) 设计各级中断处理程序的中断屏蔽字（假设 1 为屏蔽，0 为开放）。
- b) 若在运行主程序时，同时出现第 2、4 级中断请求，而在处理第 2 级中断过程中，又同时出现 1、3、5 级中断请求，画出此程序运行过程示意图。
- a) 1 级中断的处理优先级最高，说明 1 级中断对其他所有中断都屏蔽，其屏蔽字为全 1；3 级中断的处理优先级最低，所以除了 3 级中断本身之外，对其他中断全都开放，其屏蔽字为 00100。以此类推，得到所有中断屏蔽字：

中断程序级别	中断屏蔽字				
	1 级	2 级	3 级	4 级	5 级
第 1 级	1	1	1	1	1
第 2 级	0	1	1	0	0
第 3 级	0	0	1	0	0
第 4 级	0	1	1	1	1
第 5 级	0	1	1	0	1

- b) 程序运行过程示意图：

在运行主程序时，同时出现中断 2 和中断 4。因为主程序对所有中断都开放，而中断 2 的响应优先级高于中断 4，所以先响应中断 2。（发现中断 2 和 4 后，会停止响应中断一段时间，即“关中断”，这段时间内会决定是响应中断 2 和 4 中的哪个中断，保存原有的状态，并在响应中断 2 之后、处理中断 2 之前“开中断”。）

在处理中断 2 之前，由于中断 4 的处理优先级高于中断 2，因此响应并处理中断 4，结束后返回中断 2 并处理。

在处理中断 2 过程中，同时发生了 1、3、5 级中断。因为中断 2 屏蔽中断 3，且中

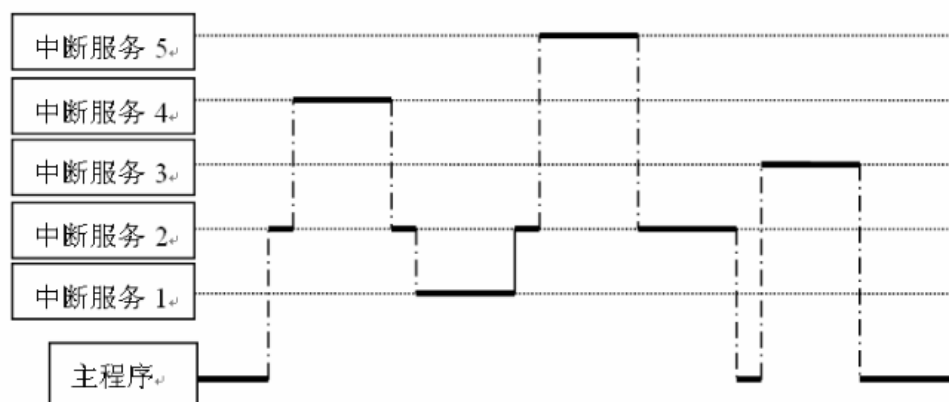


断 1 的响应优先级高于中断 5，所以响应中断 1。因为中断 1 的处理优先级最高，所以在其处理过程中不会响应任何新的中断请求，直到中断 1 处理结束，然后返回中断 2。

由于中断 5 的优先级高于中断 2，所以会响应并处理中断 5，处理完后回到中断 2。

由于中断 2 能够屏蔽中断 3，所以中断 2 会处理完，并回到主程序。

主程序响应并处理中断 3，处理结束后回到主程序。



31. 题目 17 中，如果设备的速度提高到 40Mbps，则上述 3 种方式中，哪些是不可行的？为什么？对于可行的方式，计算出 CPU 花在该设备 I/O 操作上的时间占整个处理器时间的百分比？

当设备的速度提高到 40Mbps 后，传送 1 个字节需要的时间为  $0.2 \mu s$ 。

查询方式中获取第 1 个字节后，第 2 次查询最多可能间隔  $(1000+60) \times 1ns =$

$1.06 \mu s > 0.2 \mu s$ ，会造成数据丢失，所以不可行。

中断方式中获取第 1 个字节后，响应第 2 次中断的时间间隔至少为  $(2+1200) \times 1ns = 1.202 \mu s > 0.2 \mu s$ ，会造成数据丢失，所以不可行。

采用周期挪用 DMA 方式，假设 DMA 与 CPU 不存在访存冲突，占用 CPU 时间为  $2000 \times 1ns = 2 \mu s$ 。如果 DMA 接收数据的速度慢于存入主存的速度，则所用的总时间为接收数据的时间加上将最后一个字节存入主存的时间再加上初始化和结束的时间；如果 DMA 接收数据的速度快于存入主存的速度，则所有的总时间为接收第一个数据的时间加上将所有数据存入主存的时间再加上初始化和结束的时间。所以占用 CPU 时间的比例为：

$2 \mu s / \max \{1000B/40Mbps + 1ns \times 1000 + 2 \mu s, 0.2 \mu s + 1000 \times 1 \mu s + 2 \mu s\} = 0.20\%$ 。

[卢苇，121250091][杨松，121250190][马昕，131250093][苏语风，181250127]

（这里与几位同学进行了讨论，主要分歧在于“DMA 处理一个字节需要 1 个主存周期”中“主存周期”的大小到底是 1 个时钟周期还是 1000 个时钟周期。最终觉得还是 1000 个时钟周期更为合理，即 CPU 与 DMA 访问主存的时间开销是一样的，区别是采用 DMA 访问主存时不需要 CPU 的参与。）