

得分

### 第三题 (20 分)

(1) 观察下面C语言函数和它相应的x86-64汇编代码

```
int foo(int x, int i)
{
    switch(i)
    {
        case 1:
            x -= 10;
        case 2:
            x *= 8;
            break;
        case 3:
            x += 5;
        case 5:
            x /= 2;
            break;
        case 0:
            x &= 1;
        default:
            x += i;
    }
    return x;
}

00000000004004a8 <foo>:
4004a8: mov %edi,%edx
4004aa: cmp $0x5,%esi
4004ad: ja 4004d4 <foo+0x2c>
4004af: mov %esi,%eax
4004b1: jmpq *0x400690(,%rax,8)
4004b8: sub $0xa,%edx
4004bb: shl $0x3,%edx
4004be: jmp 4004d6 <foo+0x2e>
4004c0: add $0x5,%edx
```

```

4004c3: mov %edx,%eax
4004c5: shr $0x1f,%eax
4004c8: lea (%rdx,%rax,1),%eax
4004cb: mov %eax,%edx
4004cd: sar %edx
4004cf: jmp 4004d6 <foo+0x2e>
4004d1: and $0x1,%edx
4004d4: add %esi,%edx
4004d6: mov %edx,%eax
4004d8: retq

```

调用gdb命令x/kg \$rsp 将会检查从rsp中的地址开始的k个8字节字，请填写下面gdb命令的输出（每空一分）。

>(gdb) x/6g 0x400690

```

0x400690: 0x_____ 0x_____
0x4006a0: 0x_____ 0x_____
0x4006b0: 0x_____ 0x_____

```

(2) 右边的汇编代码是由左边程序中的 m 函数编译而成。回答如下问题。

<pre> typedef struct _list {     struct _list* next;     int value; } list;  int m(list* p) {     int r = 100;     while ( __①__ ) {         r = __②__;         p = __③__;     }     if (p != 0) r = __④__;     return r; } </pre>	<pre> m:     testq %rdi, %rdi     je .L6     movq (%rdi), %rdx     movl \$100, %eax     testq %rdx, %rdx     jne .L4     jmp .L3 .L14:     movq (%rdi), %rdx     testq %rdx, %rdx     je .L3 .L4:     movl 8(%rdx), %r8d     __⑤__ 8(%rdi), %r8d     __⑤__ %r8d, %eax     movq (%rdx), %rdi     testq %rdi, %rdi     jne .L14     ret .L3:     __⑤__ 8(%rdi), %eax     ret .L6:     movl \$100, %eax     ret </pre>
--	---

已知访存延迟 1 个指令周期。不访存的时候 addl 延迟 4 个指令周期，imull 延

迟 8 个指令周期，其他指令延迟 1 个指令周期。指令访存时延迟的执行周期为不访存的时候延迟的指令周期和访存延迟的周期之和。函数 m 中的循环在处理链表 p 的时候 CPE 为 4 个指令周期。⑤处的指令为 `movl`, `addl`, `imull` 中的一个。请填写①-⑤处的代码（⑤空 2 分，其他 3 分）。

- ① \_\_\_\_\_
- ② \_\_\_\_\_
- ③ \_\_\_\_\_
- ④ \_\_\_\_\_
- ⑤ \_\_\_\_\_

得分

第四题（20 分）

（出题人：汪小林，易江芳）

请分析32位的Y86 ISA中新加入的一组条件返回指令：cretXX，其格式如下。

cretXX	9	fun
--------	---	-----

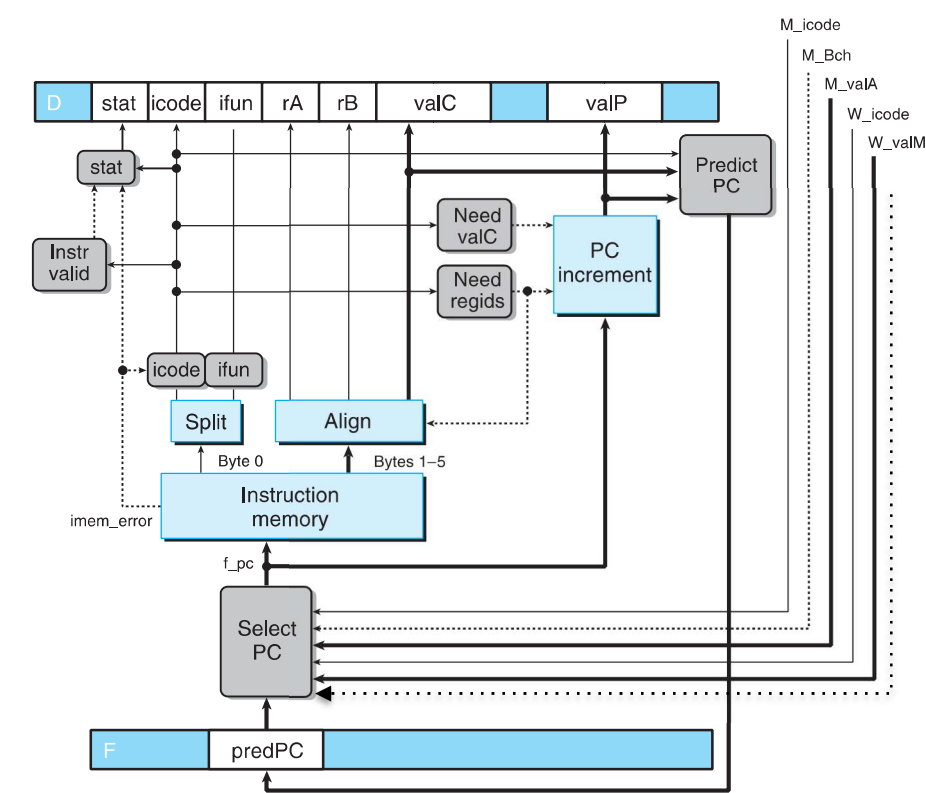
类似cmovXX，该组指令只有当条件码 (Cnd) 满足时，才执行函数返回；如果条件不满足，则顺序执行。

1. 若在教材所描述的SEQ处理器上执行这条指令，请按下表补全每个阶段的操作。需说明的信号可能会包括：icode, ifun, rA, rB, valA, valB, valC, valE, valP, Cnd; the register file R[], data memory M[], Program counter PC, condition codes CC。其中对存储器的引用必须标明字节数。如果在某一阶段没有任何操作，请填写none指明。

Stage	cretXX Offset
Fetch	_____
Decode	
Execute	
Memory	_____
Write back	
PC update	_____

2. 为了执行cretXX指令，我们需要改进教材所描述的PIPE处理器，在W（Write

Back) 阶段引入流水线寄存器\_\_\_\_\_，并将其连接到PC选择器 (Select PC) 以便有条件地更新PC。假设改进后的处理器总是预测函数返回条件不满足，则如果返回条件满足时，一共会错误取指\_\_\_\_\_条指令。



3. 在2中改进的PIPE处理器上执行cretXX指令时，发生预测错误时的判断条件和各级流水线寄存器的控制信号应如何设置？

Condition	Trigger
Mispredicted cret	( <u>          = ICRETXX &amp;&amp;          </u> )    ( <u>          = ICRETXX &amp;&amp;          </u> )

Condition	F	D	E	M	W
Mispredicted cret				normal	normal

4. PIPE 处理器上处理器上执行如下代码片段,

```
0x000:    xorl %eax, %eax
```

```
0x002:    popl %esp
```

```
0x004:    cretne
```

是否会发生 load-use 和 misprediction cret 组合的 hazard 情况?

如果此时 “popl %esp” 在流水线的 Execute 阶段, 请问此时, 各级流水线寄

Condition	F	D	E	M	W
Combination				normal	normal

得分

第五题（20 分）

现有一个能够存储4个Block的Cache，每一个Cache Block的长度为2 Byte（既  $B = 2$ ）。内存空间的大小是32Byte，既内存空间地址范围如下：

$0_{10}$  ( $00000_2$ ) --  $31_{10}$  ( $11111_2$ )

现有一程序，访问内存地址序列如下所示，单位是Byte。

$2_{10}$   $23_{10}$   $10_{10}$   $9_{10}$   $9_{10}$   $11_{10}$   $3_{10}$

1. Cache的结构如下图所示（ $S=2$ ， $E=2$ ），初始状态为空，替换策略LRU。请在下图空白处填入上述数据访问后Cache的状态。

（TAG使用二进制格式；Data Block使用10进制格式，例：M[6-7]表示地址 $6_{10}$ - $7_{10}$ 对应的数据）

	V	TAG	Data Block
set0			

	V	TAG	Block
set1			

上述数据访问一共产生了多少次 Hit：\_\_\_\_\_（2pts）

2. 如果Cache的替换策略改成MRU（即Most Recently Used，最近使用的数据被替换出去），请在下图空白处填入访问上述数据访问后Cache的状态。

V	TAG	Data Block
---	-----	------------

set0			
set1	V	TAG	Block

上述数据访问一共产生了多少次 Hit: \_\_\_\_\_ (2pts)

3. 现增加一条新规则：地址区间包含 5 的倍数的 block 将不会被缓存，仍旧使用 MRU 替换策略，这 7 次数据访问一共产生了多少次 Hit \_\_\_\_\_ (2pts)

4. 在第 3 小题的基础上，现又增加一条数据预取规则：每当地址为 10 的数据被访问时，地址为 8 的数据将会被放入缓存，这 7 次数据访问一共产生了多少次 Hit \_\_\_\_\_ (2pts)