

得分

第四题（20 分）

分析32位的Y86 ISA中新加入的条件内存传送指令：`crmmovqXX`和`cmrmovqXX`。  
`crmmovqXX`和`cmrmovqXX`指令在条件码满足所需要的约束时，分别执行和  
`rmmovq`以及`mrmovq`同样的语义。其格式如下：

<code>rmmovq</code>	4	0	<code>rA</code>	<code>rB</code>	D (8字节)
<code>crmmovqXX</code>	4	<code>fn</code>	<code>rA</code>	<code>rB</code>	D (8字节)
<code>mrmovq</code>	5	0	<code>rA</code>	<code>rB</code>	D (8字节)
<code>cmrmovqXX</code>	5	<code>fn</code>	<code>rA</code>	<code>rB</code>	D (8字节)

1. 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`; 寄存器堆`R[]`, 存储器`M[]`, 程序计数器`PC`, 条件码`CC`。其中对存储器的引用必须标明字节数。

阶段	<code>rmmovq rA, D(rB)</code>	<code>cmrmovqXX D(rB), rA</code>
取指		
译码	$valA \leftarrow R[rA]$ $valB \leftarrow R[rB]$	
执行		
访存		
写回	none	
更新PC	$PC \leftarrow valP$	

2. 为了执行上述新增指令，我们需要改进教材所描述的PIPE处理器，在回写（W: Write Back）阶段引入寄存器以保持流水线信号\_\_\_\_\_（请参考教材对信号的命名规则书写），以便有条件地更新寄存器内容。在如此改进的PIPE处理器上，请写出如下信号的HCL代码。

信号	HCL代码
F_stall	(E_icode in {IMRMVQ, IPOPOQ}    ( _____ ① )) && E_dstM in _____ ②    IRET in _____ ③
E_bubble	( _____ ④ )    (E_icode in {IMRMVQ, IPOPOQ}    ( _____ ① )) && E_dstM in _____ ②
M_bubble	m_stat in _____ ⑤    W_stat in _____ ⑤

附HCL描述中的常数值编码表如下：

IHALT	halt指令的代码	INOP	nop指令的代码
IRRMVQ	rrmovq指令的代码	IIRMOVQ	irmovq指令的代码
IRMMOVQ	rmmovq指令的代码	IMRMVQ	rmrmovq指令的代码
ICRMMOVQ	crmmovqXX指令的代码	ICMRMOVQ	cmrmovqXX指令的代码
IOPL	整数运算指令的代码	IJXX	跳转指令的代码
ICALL	call指令的代码	IRET	ret指令的代码
IPUSHQ	pushq指令的代码	IPOPOQ	popq指令的代码
FNONE	默认功能码	RNONE	表示没有寄存器文件访问
ALUADD	表示加法运算	RRSP	表示%rsp寄存器ID
SAOK	正常地址操作状态码	SADR	地址异常状态码
SINS	非法指令异常状态码	SHLT	halt状态码

3. 对于下面的Y86汇编代码, 请使用上述条件内存传送指令将其修改为不带跳转的汇编代码序列。假设下面的代码片段在教材所描述的PIPE处理器上运行, 不考虑该片段前后代码的影响以及高速缓存 (cache) 失效的情况, 假设%rsi初值为0, 处理器设计使用总是选择 (always taken) 的预测策略。原始代码片段预计运行\_\_\_\_\_周期, 改进代码片段预计执行\_\_\_\_\_周期。

原始代码	改进代码
<pre> andq %rsi, %rsi jne L1 mrmovq 8(%rdx), %rax j L2 L1: mrmovq 8(%rdx), %rbx L2: addq %rax, %rbx </pre>	

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 的大小为 2 Byte（即  $B = 2$ ）。内存空间的大小是 32 Byte，即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。

$0_{10}$   $3_{10}$   $4_{10}$   $7_{10}$   $16_{10}$   $19_{10}$   $21_{10}$   $22_{10}$   $8_{10}$   $10_{10}$   $13_{10}$   $14_{10}$   $24_{10}$   $26_{10}$   $29_{10}$   $30_{10}$

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 (Least Recently Used, 最近最少使用)。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 使用十进制格式，例：M[6-7] 表示地址  $6_{10}-7_{10}$  对应的数据)

	V	TAG	Data Block	V	TAG	Data Block
set0						
set1						

上述数据访问一共产生了多少次 Hit：\_\_\_\_\_

2. 在第 1 小题的基础上，现增加一条数据预取规则：每当 cache 访问出现 miss 时，被访问地址及其后续的一个 cache block 都会被放入缓存，即当 M[0-1] 访问发生 miss，则把 M[0-1] 和 M[2-3] 都放入缓存中。那么，这 16 次数据访问一共产生了多少次 Hit：\_\_\_\_\_

3. 在第 1 小题的基础上，如果每一个 Cache Block 的大小扩大为 4 Byte（即  $B = 4$ , cache 大小变为原来的 2 倍），这 16 次数据访问一共产生了多少次 Hit：\_\_\_\_\_

4. 在第 3 小题的基础上，考虑增加 2 中的数据预取规则，这 16 次数据访问一共产生了多少次 Hit：\_\_\_\_\_