

第5章 布置作业

- 课后4、5、6、7、8、9、10、13、14、15

P191 第4题 参考答案

采用同步方式读写内存，CALL指令有2个字，按字编址，每次从主存读取一个字，CALL指令需要读两次主存，一次是读取指令中的操作码，第二次是读取指令中的给出的子程序首地址，指令周期分为以下三个阶段：

(1) 读指令操作码：将PC的内容作为地址访问存储器，取出指令的操作码，送IR，同时PC+1送PC，以指向指令的第二个字。至少需要三个时钟周期。

PCout, MOVb, MARin

Read, b+1, PCin

MDRout, MOVb, IRin

(2) 取子程序首地址：将PC的内容作为地址，取出指令的第二个字(即子程序入口地址)送PC，以使下一个指令周期从子程序的第一条指令开始执行。同时计算PC+1以得到返回地址，送Y寄存器(临时保存)。至少需要三个时钟周期。

PCout, MOVb, MARin

Read, b+1, Yin

MDRout, MOVb, PCin

(3) 保存返回地址至栈中：将临时保存在Y寄存器的返回地址送到栈顶保存，并自动调整栈顶指针。至少需要三个时钟周期。

SPout, MOVb, MARin

Yout, MOVb, MDRin

Write, SPout, b-1, Spin

备注：也可以将第一次PC+1的结果送到Y,第二阶段以Y访问主存，继续Y+1,送PC，也能实现Call，都是9个时钟周期。

P191 第5题 参考答案

图5.7所示的数据通路中，所有与内部总线相连的寄存器都有相应的Rin或Rout控制信号，以控制总线和寄存器之间的数据传送。总线和ALU输入端之间、Y寄存器与ALU输入端之间都无需控制信号。下面给出的是指令执行阶段的控制信号，此时指令的第二个字(Imm16)已从存储器取出并被存放在MDR中。

(1) 指令功能为： $R[R1] \leftarrow R[R1] + \text{Imm16}$ ，执行阶段不需要访存操作，因此可用以下三个时钟周期完成。

MDRout, Yin

R1out, add, Zin

Zout, R1in

(2) 指令功能为： $R[R1] \leftarrow R[R1] + M[\text{Imm16}]$ ，执行阶段需要一次访存操作，因此至少需要以下5个时钟周期完成。

MDRout, MARin

Read1, (R1out, Yin) //R1out, Yin与Read1同时送出，并在Read2阶段保持不变，也可以与Read2同时送出。

Read2, R1out, Yin //同步等待，再保持一个时钟周期，等待存储器

MDRout, add, Zin

Zout, R1in

P191 第5题 参考答案

图5.7所示的数据通路中，所有与内部总线相连的寄存器都有相应的Rin或Rout控制信号，以控制总线和寄存器之间的数据传送。总线和ALU输入端之间、Y寄存器与ALU输入端之间都无需控制信号。下面给出的是指令执行阶段的控制信号，此时指令的第二个字(Immm16)已从存储器取出并被存放在MDR中。

(3) 指令功能为： $R[R1] \leftarrow R[R1] + M[M[Imm16]]$ ，执行阶段需要2次访存操作，因此至少需要以下8个时钟周期完成。

MDRout, MARin

Read1

Read2

MDRout, MARin

Read1, (R1out, Yin)

Read2, R1out, Yin

MDRout, add, Zin

Zout, R1in

P192 第6题 参考答案

假定图5.22单周期数据通路对应的控制逻辑发生错误，使得控制信号RegWr、RegDst、Branch、MemWr、ExtOp、R-type中某一个在任何情况下总是为0，则该控制信号为0时，哪些指令不能正确执行？要求分别讨论。

若RegWr=0，则所有需写结果到寄存器的指令（例如R-型指令、load类指令等）都不能正确执行，因为寄存器不发生写操作。

若RegDst=0，则所有R-型指令都不能正确执行，因为目的寄存器指定为Rt，而不是Rd。

若Branch=0，则branch类指令可能出错，因为永远不会发生转移。

若MemWr=0，则Store类指令不能正确执行，因为存储器不能写入所需数据。

若ExtOp=0，则需要符号扩展的指令(例如beq、lw/sw等)发生错误，因为进行的是零扩展。

若R-type=0，则所有R-type类指令执行可能出错，因为控制信号ALUctr的取值就不再是，来自对R-型指令进行解释的ALU局部控制器。

P192 第7题 参考答案

假定图5.22单周期数据通路对应的控制逻辑发生错误，使得控制信号RegWr、RegDst、Branch、MemWr、ExtOp、R-type中某一个在任何情况下总是为1，则该控制信号为1时，哪些指令不能正确执行？要求分别讨论。

若RegWr=1，则不需写结果到寄存器的指令（例如sw指令、beq指令等）都不能正确执行，因为错误地将某个值写入了某个寄存器。

若RegDst=1，则lw等部分I-型指令都不能正确执行，因为目的寄存器指定为Rd，而不是Rt。

若Branch=1，则非branch类指令可能出错，因为可能会发生不必要的转移。

若MemWr=1，则除Store类外其他指令都不能正确执行，因为会写入数据到存储器。

若ExtOp=1，则需要零扩展的指令(例如ori等)会发生错误，因为进行的是符号扩展。

若R-type=1，则图5.28可以看出R-type=1则非R-type类指令不能正确执行。

P192 第8题 参考答案

(1) 写出MIPS中用伪指令方式实现“swap rs, rt”时的指令序列
要求不能使用额外的寄存器，如下：

```
xor rs,rs,rt  
xor rt,rs,rt  
xor rs,rs,rt           //把xor 改为add?
```

那这样呢？

```
add $rtemp, $rs, $zero  
add $rs, $rt, $zero  
add $rt, $rtemp, $zero
```

(2) 假定“swap rs, rt”指令所占比例为 x ($0 \leq x \leq 1$)，其他指令比例为 $1-x$ 。则使用硬件实现该指令时，程序执行时间为原来的 $1.1(x+1-x)=1.1$ 倍。用软件实现该指令时，程序执行时间为原来的 $3x+1-x=2x+1$ 倍。因此当 $1.1 < 2x+1$ 时，硬件实现才有意义，由此可知， $x > 0.05$ ，也即：当“swap rs,rt”指令在程序中的比例大于5%时，才值得用硬件方式来实现该指令。

P192 第9题 参考答案

假定图5.31多周期数据通路对应的控制逻辑发生错误，使得控制信号PCWr、MemtoReg、IRWr、RegWr、BrWr、MemWr、PCWrCond、R-type中某一个在任何情况下总是为0，则该控制信号为0时，哪些指令不能正确执行？要求分别讨论。

若PCWr=0，则所有指令都不能正确执行，因为无法正确地更新PC。

若MemtoReg=0，则所有load类指令都不能正确执行，因为写入寄存器的是ALU输出，而不是读出的存储单元的内容。

若IRWr=0，则所有指令都不能正确执行，因为IR中不能写入当前指令。

若RegWr=0，则所有需要写结果到寄存器的指令都不能正确执行(例如R-type类指令、lw等)都不能正确执行，因为寄存器操作不能发生写操作。

若BrWr=0，则分支目标地址不能计算写入分支目标地址寄存器，则分支指令不能执行(例如beq指令)。

若MemWr=0，则Store类指令不能正确执行，因为存储器不能写入所需数据。

若PCWrCond=0，则branch类指令执行可能出错，因为永远不会转移。

若R-type=0，则所有R-type类指令执行可能出错，因为控制信号ALUctr的取值就不再是，来自对R-型指令进行解释的ALU局部控制器。

P192 第10题 参考答案

假定图5.31多周期数据通路对应的控制逻辑发生错误，使得控制信号PCWr、MemtoReg、IRWr、RegWr、BrWr、MemWr、PCWrCond、R-type中某一个在任何情况下总是为1，则该控制信号为1时，哪些指令不能正确执行？要求分别讨论。

若PCWr=1，则程序顺序执行失控，因为每个时钟都会更新PC。

若MemtoReg=1，则除了load类，所有指令都不能正确执行，因为选择了错误的信息写入寄存器。

若IRWr=1，则所有指令都可能出错，因为每个时钟周期都会写入IR，因此写入的不是当前指令。

若RegWr=1，则所有不需要写结果到寄存器的指令都不能正确执行(例如sw指令、beq)都不能正确执行，因为错误地将某个值写入了某个寄存器。

若BrWr=1，则非分支指令不能执行。

若MemWr=1，则Store类之外的指令不能正确执行，因为存储器写入了不该写的信息。

若PCWrCond=1，则除branch类外的所有指令执行可能出错，因为可能会发生不必要的转移。

若R-type=1，则所有非R-type类指令执行可能出错，因为控制信号ALUctr的取值是来自对R-型指令进行解释的ALU局部控制器。

P192 第13题 参考答案

对于某个MIPS多周期处理器，假定将访问数据的过程分成两个时钟周期，则可使时钟频率从4.8GHz提高到5.6GHz，但这样会使得lw和sw指令增加时钟周期数。已知基准程序CPUint 2000中各类执行的频率为：Load-25%, Store-10%, Branch-11%, Jump-2%, ALU-52%。那么以基准程序CPUint 2000为标准，处理器时钟频率提高后的性能提高了多少？若将取指令过程再分成2个时钟周期，则可进一步使时钟频率提高到6.4GHz，此时，时钟频率的提高是否也能带来处理器性能的提高？为什么？

假设M1、M2和M3分别表示时钟频率为4.8GHz、5.6GHz和6.4GHz的多周期处理器，对课本中的图5.32进行分析得知：Load, Store, Branch, Jump和ALU类指令在M1中的CPI分别为5, 4, 3, 3, 4，在M2中的CPI分别为6, 5, 3, 3, 4，在M3中的CPI分别为7, 6, 4, 4, 5。因此，用基准程序CPUint 2000来计算，得到它们的综合CPI如下：

$$\text{CPI}(\text{M1}) = 25\% * 5 + 10\% * 4 + 11\% * 3 + 2\% * 3 + 52\% * 4 = 4.12$$

$$\text{CPI}(\text{M2}) = 25\% * 6 + 10\% * 5 + 11\% * 3 + 2\% * 3 + 52\% * 4 = 4.47$$

$$\text{CPI}(\text{M3}) = 25\% * 7 + 10\% * 6 + 11\% * 4 + 2\% * 4 + 52\% * 5 = 5.47$$

因此，M1、M2和M3的MIPS数分别如下：

$$\text{MIPS}(\text{M1}) = 4800 / 4.12 \approx 1165$$

$$\text{MIPS}(\text{M2}) = 5600 / 4.47 \approx 1253$$

$$\text{MIPS}(\text{M3}) = 6400 / 5.47 \approx 1170$$

由此可见，数据访问改为双周期的做法效果好，进一步把取指令改为双周期的做法反而使得MIPS变小了不可取。因为数据访问只涉及lw/sw指令，而取指令涉及所有指令，使得CPI显著增大，从而降低了性能。

P192 第14题 参考答案

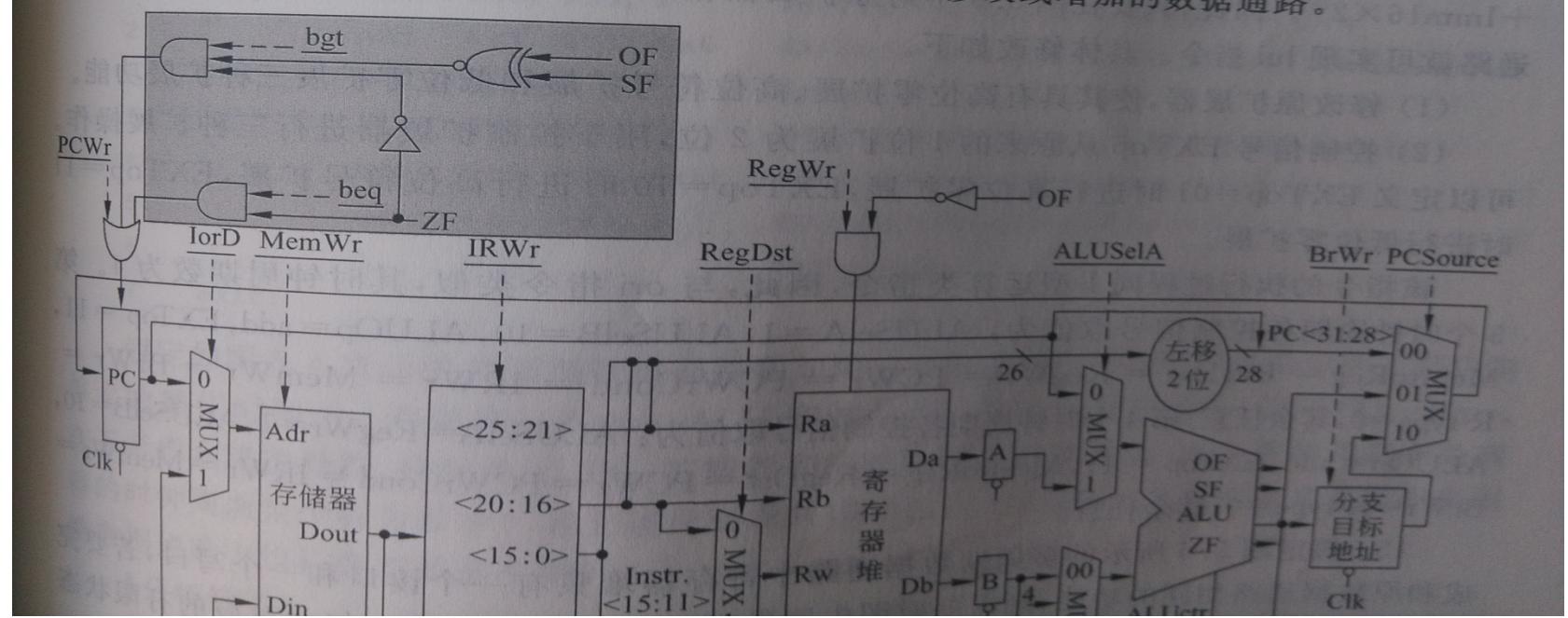
OF=SF且ZF=0时，表示大于。

对于两个有符号数比较(有符号数相减考虑SF和OF): CMP AL,BL，分两种情况:

1.两数同为正数或负数（只看SF，此时OF=0）——SF=0，则AL>BL；SF=1，则AL<BL

2.两数一正一负（比较SF与OF）——SF=OF，则AL>BL；SF不等于OF，则AL<BL

比较指令需要对两个寄存器 rs 和 rt 中的内容做减法运算(用 ALU 中的加法器实现)，ALU 根据运算结果得到 ZF、SF 和 OF 三个标志信息。在此，rs 和 rt 中的内容被看成带符号整数，因此，判断 $R[rs] > R[rt]$ 的条件表达式为 $\overline{ZF} \cdot (SF \oplus OF)$ 。图 5.6 中的多周期数据通路已能支持 beq 指令的执行，而 beq 指令和 bgt 指令的目标转移地址是一样的，因此，只要在图 5.6 中原先 beq 指令的条件判断逻辑基础上增加 bgt 指令的条件判断逻辑即可，修改后的数据通路如图 5.13 所示，其中带阴影部分是修改或增加的数据通路。



P192 第15题 参考答案

微程序控制器容量为 1024×48 位，说明微地址占10位，微指令字为48位。微指令字分为三个字段：微命令字段、转移控制字段和下地址子字段。因为微地址10位，所以下地址字段10位，用来给出下条微指令的地址；因为需要对5种情况进行控制，所以控制转移字段有3位，剩下的 $48 - 10 - 3 = 35$ 位是微命令字段。

采用断定法时，转移控制字段可按下方案进行编码：

- 000：下地址字段指出的地址作为下条微地址。
- 100：根据分支1处的条件来选择下条微地址。
- 101：根据分支2处的条件来选择下条微地址。
- 110：根据分支3处的条件来选择下条微地址。
- 111：根据分支4处的条件来选择下条微地址。

微指令的格式如下：

