

3. 在书中图6.9中，假定总线传输延迟和ALU运算时间分别是20ps和200ps，寄存器建立时间为10ps，寄存器保持时间为5ps，寄存器的锁存延迟（Clk-to-Q time）为4ps，控制信号的生成延迟（Clk-to-signal time）为7ps，三态门接通时间为3ps，则从当前时钟到达开始算起，完成以下操作的最短时间是多少？各需要几个时钟周期？

(1) 将数据从一个寄存器传送到另一个寄存器

(2) 将程序计数器PC加1

参考答案：

(1) 寄存器的锁存延迟与控制信号的生成延迟的时间重叠，
且 $\text{Clk-to-signal time} > \text{Clk-to-Q time}$ ，所以完成寄存器传送的时间延迟为：

$$7+3+20+10=40\text{ps}。$$

因为在这个过程中，只要最后保存一次信息到寄存器，所以只需要一个时钟周期。

(2) 分两个阶段：

$$\text{PC}+1 \rightarrow \text{Z} : 7+3+20+200+10=240\text{ps};$$

$$\text{Z} \rightarrow \text{PC} : 7+3+20+10=40\text{ps}$$

寄存器保持时间用来作为时间约束。

因为在这个过程中，需要经过两次总线传输，每次都需将传输信息保存在某个寄存器中，所以需要两个时钟周期。

4. 右图6.30给出了某CPU内部结构的一部分，MAR和MDR直接连到存储器总线（图中省略）。在两个总线之间的所有数据传送都需经过算术逻辑部件ALU。ALU可实现的部分功能及其控制信号如下：

MOVa: F=A; MOVb: F=B;

a+1: F=A+1; b+1: F=B+1

a-1: F=A-1; b-1: F=B-1

其中A和B是ALU的输入，F是ALU的输出。假定JSR（转子指令）指令占两个字，第一个字是操作码，第二个字给出子程序的起始地址，返回地址保存在主存的栈中，用SP（栈指示器）指向栈顶，按字编址，每次从主存读取一个字。请写出读取并执行JSR指令所要求的控制信号序列（提示：当前指令地址在PC中）。

参考 P236~239 （5.单总线数据通路（1）（2）（3））

参考答案：

假定采用同步方式（若为异步，则只需在read和Write后加一个等待信号WMFC）

分三个阶段：

1. 取指令操作码：PCout, MOVb, MARin

Read, b+1, PCin

MDRout, MOVb, IRin

2. 取子程序首址：PCout, MOVb, MARin

Read, b+1, Yin

MDRout, MOVb, PCin

3. 保存返址至栈：SPout, MOVb, MARin

Yout, MOVb, MDRin

Write, SPout, b-1, SPin

（注：若按最长的存储访问时间作为CPU时钟周期，则上述每个阶段都需三个时钟周期）
能否用更少的时钟周期完成上述功能？不能！以下是另一种方式

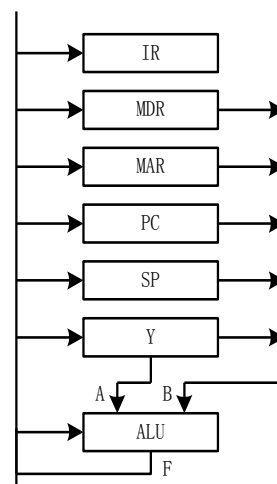


图30

1. 取指令操作码: PCout, MOVb, MARin

Read, b+1, Yin

MDRout, MOVb, IRin

2. 取子程序首址: Yout, MOVb, MARin

Read, a+1, Yin (用b+1也行)

MDRout, MOVb, PCin

3. 保存返址至栈: SPout, MOVb, MARin

Yout, MOVb, MDRin

Write, SPout, b-1, SPin

5. 假定某计算机字长16位, CPU内部结构如书中图6.9所示, CPU和存储器之间采用同步方式通信, 按字编址。采用定长指令字格式, 指令由两个字组成, 第一个字指明操作码和寻址方式, 第二个字包含立即数Imm16。若一次存储访问所花时间为2个CPU时钟周期, 每次存储访问存取一个字, 取指令阶段第二次访存将Imm16取到MDR中, 请写出下列指令在指令执行阶段的控制信号序列, 并说明需要几个时钟周期。

(1) 将立即数Imm16加到寄存器R1中, 此时, Imm16为立即操作数。

即: $R[R1] \leftarrow R[R1] + Imm16$

(2) 将地址为Imm16的存储单元的内容加到寄存器R1中, 此时, Imm16为直接地址。

即: $R[R1] \leftarrow R[R1] + M[Imm16]$

(3) 将存储单元Imm16的内容作为地址所指的存储单元的内容加到寄存器R1中。此时, Imm16为间接地址。即: $R[R1] \leftarrow R[R1] + M[M[Imm16]]$

参考 P236~239 (5、单总线数据通路 (1) (2) (3))

参考答案:

(1) MDRout, Yin

R1out, add, Zin

Zout, R1in

需3个时钟周期

(2) MDRout, MARin

Read1, (R1out, Yin也可以放在该控制信号所在的时钟周期中)

Read2, R1out, Yin

MDRout, add, Zin

Zout, R1in

需5个时钟周期

(3) MDRout, MARin

Read1

Read2

MDRout, MARin

Read1, (R1out, Yin)

Read2, R1out, Yin

MDRout, add, Zin

Zout, R1in

需8个时钟周期

6. 假定图6.24单周期数据通路对应的控制逻辑发生错误, 使得在任何情况下控制信号 RegWr、RegDst、Branch、MemWr、ExtOp、R-type总是为0, 则哪些指令不能正确执行? 为什么?

参考答案:

若 $\text{RegWr} \equiv 0$ ，则所有需写结果到寄存器的指令（如：R-Type指令、load指令等）都不能正确执行，因为寄存器不发生写操作；
 若 $\text{RegDst} \equiv 0$ ，则所有R-Type指令都不能正确执行，因为目的寄存器指定错误；
 若 $\text{Branch} \equiv 0$ ，则Branch指令可能出错，因为永远不会发生转移；
 若 $\text{MemWr} \equiv 0$ ，则Store指令不能正确执行，因为存储器不能写入所需数据；
 若 $\text{ExtOp} \equiv 0$ ，则需要符号扩展的指令（如Beq、lw/sw等）发生错误。
 若 $\text{R-type} \equiv 0$ ，则所有R-type指令的执行可能出错

7. 假定图6.24单周期数据通路对应的控制逻辑发生错误，使得在任何情况下控制信号 RegWr 、 RegDst 、 Branch 、 MemWr 、 ExtOp 、 R-type 总是为1，则哪些指令不能正确执行？为什么？

参考 P246~251 6.2.2-3,4, 5,6,7

参考答案：

若 $\text{RegWr} \equiv 1$ ，则所有不需写结果到寄存器的指令（如：sw、beq等）都不能正确执行；
 若 $\text{Regdst} \equiv 1$ ，则lw和ori等指令不能正确执行，因为目的寄存器指定错误；
 若 $\text{Branch} \equiv 1$ ，则非Branch指令可能出错，因为可能会发生不必要的转移；
 若 $\text{MemWr} \equiv 1$ ，则除Store指令外其他指令都不能正确执行，因为存储器总会写入数据；
 若 $\text{ExtOp} \equiv 1$ ，则需要零扩展的指令（如ori等）会发生错误。
 若 $\text{R-type} \equiv 1$ ，则所有非R-type指令的执行可能出错

8. 在MIPS指令集中需要增加一条swap指令，可以使用软件方式用若干条已有指令来实现伪指令，也可以通过改动硬件来实现。

（1）写出用伪指令方式实现“swap \$rs, \$rt”时的指令序列

（2）假定用硬件实现时会使一条指令的执行时间增加10%，则swap指令在程序中占多大的比例才值得用硬件方式来实现？

参考答案：

（1） swap指令可用以下三条指令实现。

```
xor $rs, $rs, $rt
xor $rt, $rs, $rt
xor $rs, $rs, $rt
```

（若使用额外寄存器\$rtmp，则\$rtmp内容会被破坏，所以伪指令一般不能用额外寄存器）

```
add $rtmp, $rs, $zero
add $rs, $rt, $zero
add $rt, $rtmp, $zero
```

（2）假定该指令占x%，其他指令占(1-x)%

则用硬件实现该指令时，程序执行时间为原来的 $1.1 \cdot (x+1-x) = 1.1$ 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

当 $1.1 < 2x+1$ 时，硬件实现才有意义

由此可知， $x > 5\%$

9. 假定图6.33多周期数据通路对应的控制逻辑发生错误，使得在任何情况下控制信号 PCWr 、 IRWr 、 RegWr 、 BrWr 、 PCSource 、 MemWr 、 MemtoReg 、 PCWrCond 、 R-type 总是为0，则哪些指令不能正确执行？为什么？

参考P260~263（6.3.2 指令执行状态分析）

参考答案：

若 $\text{PCWr} \equiv 0$ ，则所有指令都不正确，因为无法更新PC

若 $IRWr=0$ ，则所有指令都不能正确执行，因为IR中不能写入指令
 若 $RegWr=0$ ，则所有需要写结果到寄存器的指令（如：R-Type指令、load指令等）都不能正确执行，因为寄存器不发生写操作
 若 $PCSource=00$ ，则除j之外的其他指令都不能正确得到下条指令地址
 若 $MemWr=0$ ，则Store指令不能正确执行，因为存储器不能写入数据
 若 $MemtoReg=0$ ，则所有Load指令执行错误，因为寄存器写入的是ALU输出
 若 $PCWrCond=0$ ，则Branch指令不能正确执行，因为不能写入转移目标地址到PC
 若 $R-type=0$ ，则所有R-type指令的执行可能出错

10. 假定P.185图6.32多周期数据通路对应的控制逻辑发生错误，使得在任何情况下控制信号PCWr、IRWr、RegWr、BrWr、PCSource、MemWr、MemtoReg、PCWrCond、R-type总是为1，则哪些指令不能正确执行？为什么？

参考答案：

若 $PCWr=1$ ，则程序执行顺序失控，因为每个时钟都会更新PC
 若 $IRWr=1$ ，则所有指令都可能不能正确执行，因为写入IR的可能不是当前指令
 若 $RegWr=1$ ，则所有不需写结果到寄存器的指令（如：sw、beq等）都不能正确执行
 若 $PCSource=01$ ，则j和Branch指令不能正确得到下条指令地址
 若 $MemWr=1$ ，则除Store指令外的所有指令都不能正确执行
 若 $MemtoReg=1$ ，则除Load外的所有指令执行错误
 若 $PCWrCond=1$ ，则除Branch外的其他指令可能不能正确执行
 若 $R-type=1$ ，则所有非R-type指令的执行可能出错

11. 假定有一条MIPS伪指令“Bcmp \$t1, \$t2, \$t3”，其功能是实现对两个主存块数据的比较，\$t1和\$t2中分别存放两个主存块的首地址，\$t3中存放数据块的长度，每个数据占四个字节，若所有数据都相等，则将0置入\$t1；否则，将第一次出现不相等时的地址分别置入\$t1和\$t2并结束比较。若\$t4和\$t5是两个空闲寄存器，请给出实现该伪指令的指令序列，并说明在类似于P.185图6.32的多周期数据通路中执行该伪指令时要用多少时钟周期。

12. 假定某计算机字长16位，标志寄存器Flag中的ZF、NF和VF分别是零、负和溢出标志，采用双字节定长指令字。假定Bgt (大于零转移) 指令的第一个字节指明操作码和寻址方式，第二个字节为偏移地址Imm8，其功能是：

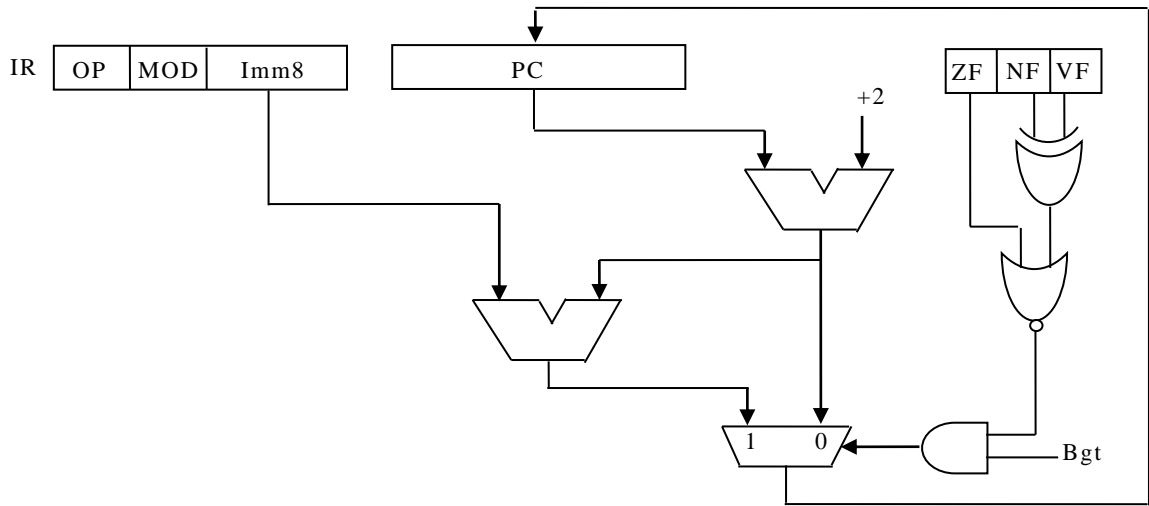
若 $(ZF+(NF \oplus VF)=0)$ 则 $PC=PC+2+Imm8$ 否则 $PC=PC+2$

- (1) 该计算机的编址单位是什么？
- (2) 画出实现Bgt指令的数据通路。

参考P248(6.分支指令的数据通路)、P232~234(6.1.4 数据通路的基本结构)、P242 (6.2.2 数据通路设计)

参考答案：

- (1) 该计算机的编址单位是字节。
- (2) 实现Bgt指令的数据通路如下



13. 对于多周期MIPS处理器，假定将访问数据的过程分成两个时钟周期可使时钟频率从4.8GHz提高到5.6GHz，但会使得lw和sw指令增加时钟周期数。已知基准程序CPUint 2000中各类指令的频率为：Load: 25%，Store: 10%，Branch: 11%，Jump: 2%，ALU: 52%。以基准程序CPUint 2000为标准计算时钟频率提高后处理器的性能提高了多少？若将取指令过程再分成两个时钟周期，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？

参考P265 例6.1 P259~P265

参考答案：

SPEC CPUint 2000混合指令频率为：

Load: 25%

Store: 10%

Branch: 11%

Jump: 2%

ALU: 52%

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，从有限状态图分析，得知：

M1中上述各类指令的CPI分别为5、4、3、3、4

M2中上述各类指令的CPI分别为6、5、3、3、4

M3中上述各类指令的CPI分别为7、6、4、4、5

$CPI_{ofM1} = 25\% \times 5 + 10\% \times 4 + 11\% \times 3 + 2\% \times 3 + 52\% \times 4 = 4.12$

$CPI_{ofM2} = 25\% \times 6 + 10\% \times 5 + 11\% \times 3 + 2\% \times 3 + 52\% \times 4 = 4.47$

$CPI_{ofM3} = 25\% \times 7 + 10\% \times 6 + 11\% \times 4 + 2\% \times 4 + 52\% \times 5 = 5.47$

（实际上就是每个指令都加一个时钟周期，所以 $CPI_{ofM3} = CPI_{ofM2} + 1 = 5.47$ ）

$MIPS_{ofM1} = 4.8G / 4.12 = 1165$

$MIPS_{ofM2} = 5.6G / 4.47 = 1253$

$MIPS_{ofM3} = 6.4G / 5.47 = 1170$

由此可见，数据存取改为双周期的做法效果较好。进一步把取指令改为双周期的做法反而使MIPS数变小了，所以不可取。因为数据存取只涉及到load/Store指令，而指令存取涉及到所有指令，使得CPI显著提高。

14. 假设MIPS指令系统中有一条I-型指令“Bgt Rs, Rt, Imm16”，其功能为：

若 $(Rs > Rt)$ 则 $PC = PC + 4 + Imm16 * 4$ 否则 $PC = PC + 4$ 。

假定ALU能产生ZF（零）、NF（符号）和VF（溢出）三个标志的输出，请在P.185图6.23所示的多周期数据通路中增加实现Bgt指令的数据通路以及相应的控制信号，并对P.194表6.9进行相应的修改，写出该指令对应的微程序。（提示：只有一条微指令，可参照beq

指令实现)

15. 微程序控制器容量为 1024×48 位，微程序可在整个控存内实现转移，反映所有指令执行状态转换的有限状态机中有4个分支点，微指令采用水平格式，微地址由专门的下地址字段确定。请设计微指令的格式，说明各字段有多少位？为什么？

参考答案：

微程序控制器容量为 1024×48 位，说明微地址占10位，微指令字共48位，其中10位下地址字段用来给出下条微地址；转移控制字段需要对5种情况进行控制，需3位。例如，

000：取指令微指令首地址

100：根据分支1处的条件选择下条微地址

101：根据分支2处的条件选择下条微地址

110：根据分支3处的条件选择下条微地址

111：根据分支4处的条件选择下条微地址

剩下的 $48 - 10 - 3 = 35$ 位用来表示微操作码字段。

(如果采用计数器法，则转移控制字段需要对6种情况进行控制，比上述5种情况多一种：即顺序执行下条微指令，此时转移控制字段也有3位。)

16. 对于多周期CPU的异常和中断处理，回答以下问题：

- (1) 对于除数为0、溢出、无效指令操作码、无效指令地址、无效数据地址、缺页、访问越权和外部中断，CPU在哪些指令的哪个时钟周期能分别检测到这些异常或中断？
- (2) 在检测到某个异常或中断后，CPU通常要完成哪些工作？简要说明CPU如何完成这些工作？
- (3) TLB缺失和cache缺失各在哪个指令的哪个时钟周期被检测到？如果检测到发生了TLB缺失和cache缺失，那么，CPU各要完成哪些工作？简要说明CPU如何完成这些工作？(提示：TLB缺失可以有软件和硬件两种处理方式。)

参考答案：

- a. “除数为0”异常在取数/译码(ID/Reg)周期进行检测
- b. “溢出”异常在R-Type指令的执行(Exe)周期进行检测
- c. “无效指令”异常在取数/译码(ID/Reg)周期进行检测
- d. “无效指令地址”、“缺页”和“访问越权”异常在取指令(IF)周期检测
- e. “无效数据地址”、“缺页”和“访问越权”异常在存储器访问(Mem)周期检测
- f. “中断”可在每条指令的最后一个周期(WB)的最后进行检测