

《计算机组成原理》

（第六讲习题答案）

厦门大学信息学院软件工程系 曾文华

2022年5月4日

第6章 中央处理器

- 6.1 中央处理器概述
- 6.2 指令周期
- 6.3 数据通路及指令操作流程
- 6.4 时序与控制
- 6.5 硬布线控制器
- 6.6 微程序控制器
- 6.7 异常与中断处理

习题 (P247-253)

- 6.2 (1) (2) (3) (4) (5) (6) (7) (8) (9) (10)
- 6.3 (1) (2) (3) (4) (5) (6) (7) (8) (9) (10) (11) (12) (13)
- 6.4
- 6.5 (1)
- 6.7 (2)
- 6.9 (3)
- 6.11
- 6.12
- 6.20
- 6.21
- 6.23

习题答案（P247-253）

• 6.2 选择题

- (1) **B**
 - **MAR、MDR、IR**：没有相应的指令对其进行操作，因此汇编语言程序员不可见
 - **PC**：有相应的指令对其进行操作，因此汇编语言程序员可见
- (2) **B**
 - 指令在取数及执行过程中需要用到：通用寄存器组、算术逻辑单元、存储器
 - 指令译码器是在取指阶段用到
- (3) **B**
 - 主存空间=4GB，字长=32位；按字节编址，需要32位地址；按字编址，则需要30位地址；因此PC的位数=30位
 - 32位定长指令格式，因此IR的位数=32位
- (4) **D**
 - A、B、C的叙述都是正确的
 - 对应单周期处理器，D的叙述是正确的；但是对应多周期处理器，D的叙述是不正确的
- (5) **A**
 - 单周期处理器必须采用专用通路结构的数据通路，不能采用单总线结构的数据通路
 - 其他叙述都是正确的

- 6.2 选择题（续）

- (6) **B**
 - 控制存储器（CS）也是按地址访问的
 - 其他叙述都是正确的
- (7) **D**
 - 硬布线控制器指令执行速度快、指令功能的修改和扩展难
- (8) **C**
 - 7个微命令需要3位、3个微命令需要2位、12个微命令需要4位、5个微命令需要3位、6个微命令需要3位，共需要15位
- (9) **C**
 - 32条指令需要： $2+4 \times 32=130$ 条微指令，因此需要8位微指令地址
- (10) **C**
 - 即使是空操作指令，其指令周期也需要完成取指令的认为，因此PC和IR寄存器的内容肯定要改变

- **6.3 简答题**

- (1) **CPU**的基本功能是什么？从实现其功能的角度分析，它应由哪些部件组成？

- 答：

- (2) **CPU**内部有哪些寄存器？功能分别是什么？哪些是程序员可见的？哪些是必须的？

- 答：

- (3) 什么是取指周期？取指周期内应完成哪些操作？

- 答：

- (4) 计算机为什么要设置时序系统？说明指令周期、机器周期和时钟周期的含义。

- 答：

- 6.3 简答题（续）

- （5）简述传统三级时序和现代时序的差异

- 答：

- （6）比较单周期MIPS处理器与多周期MIPS处理器的差异。

- 答：

- （7）组合逻辑控制器和微程序控制器各有什么特点？

- 答：

- （8）说明程序与微程序、指令与微指令的异同。

- 答：

- （9）微命令有哪几种编码方法？它们是如何实现的？

- 答：

- （10）简述微程序控制器和硬布线控制器的设计方法。

- 答：

- **6.3 简答题（续）**

- **（11）简述CPU中内部异常和外部中断的区别。**

- 答：

- **（12）简述异常与中断处理的一般流程。**

- 答：

- **（13）要支持异常与中断处理，CPU需要对硬、软件进行哪些扩展？**

- 答：

- 6.4

- 答:

- (1) a: MDR寄存器 (DR寄存器); b: IR寄存器; c: MAR寄存器 (AR寄存器); d: PC寄存器
- (2) 指令 “LDA addr” 的数据通路:
- 取指阶段的数据通路: PC -> MAR -> 主存M -> MDR -> IR, PC +1 -> PC
- 执行阶段的数据通路: IR(A) -> MAR -> 主存M -> MDR -> AC

- 6.5



- 答:

— (1)

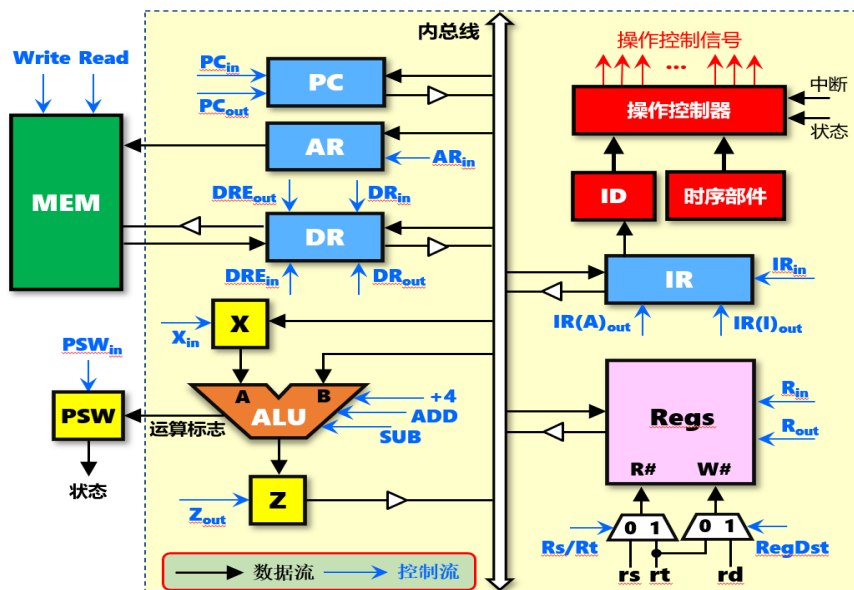
- sll为R型指令，其功能为逻辑左移，sll rd,rt,shamt $R[rd] = R[rt] \ll \text{shamt}$

- 需要增加:

- (1) IR增加IR(shamt)out控制信号，该控制信号将IR中的shamt字段输出到内总线
- (2) 运算器ALU增加SLL控制信号，该控制信号使ALU进行逻辑左移操作

- 该指令的执行周期流程为:

- T1: $R[rt] \rightarrow X$ 控制信号: $R_s/R_t=1, R_{out}=1, X_{in}=1$
- T2: $R[rt] \ll \text{shamt} \rightarrow Z$ 控制信号: $IR(\text{shamt})_{out}=1, SLL=1$
- T3: $Z \rightarrow R[rd]$ 控制信号: $Z_{out}=1, RegDst=1, R_{in}=1$



• 答:

- lui为I型指令，其功能为加载立即数，lui rt,imm R[rt]=imm<<16

- **ALU需要增加功能:** $ALUResult = SrcB$

- PC+4 (经过2个MUX) -> PC -> 指令存储器 (其中的imm) -> Sign Extend (经过MUX, 控制信号ALUSrc=1) -> SrcB -> ALU (ALU的控制信号为: $\text{AluOP}_{\text{SrcB} \rightarrow \text{ALUResult}}=1$) -> ALUResult -> (经过MUX) -> 寄存器堆的WD (控制信号RegWrite=1)

- 控制信号: $ALUSrc=1$, $AluOP_{SrcB \rightarrow ALUResult}=1$ (表示 $ALUResult=SrcB$), $RegWrite=1$

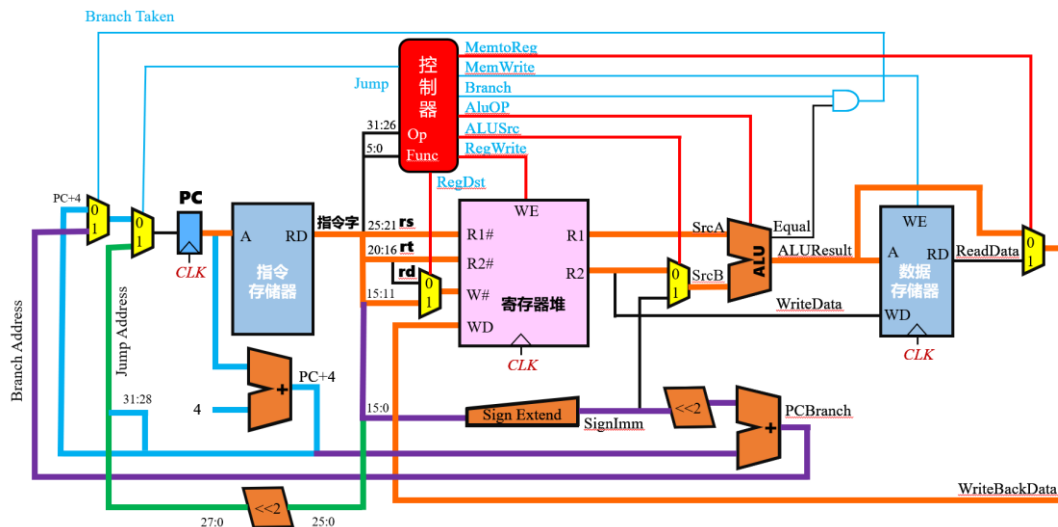


图6.25 单周期MIPS处理器的数据通路

• 6.9

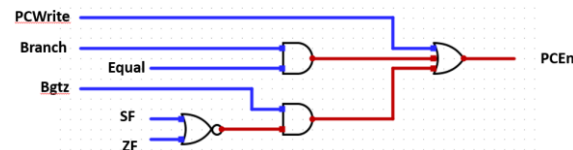
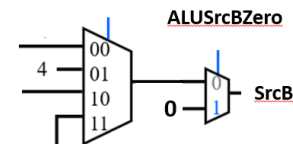
• 答:

— (3)

• **bgtz**为I型指令，其功能为大于零则转移，**bgtz** *rs*,*imm* $\text{if}(R[rs] > 0) \text{ PC} + 4 + \text{SignExt}(\text{imm}) \ll 2 \rightarrow \text{PC}$

• 需要增加:

- (1) ALU增加减法运算的功能，控制信号=SUB
- (2) ALU的标志位: SF (符号标志) 和ZF (零标志)
- (3) 4选1MUX的输出接1个2选1MUX，该MUX的另一个输入为0
- (4) 控制器增加**bgtz**指令译码输出信号**Bgtz**
- (5) ALU的标志位SF和ZF接或非门 (SF=0表示 ≥ 0 ，ZF=0表示 $\neq 0$ ；即 > 0)，再与控制器的**Bgtz**信号进行与



• 该指令的执行周期流程为:

- 取指阶段T1: $M[\text{PC}] \rightarrow \text{IR}$ $\text{PC} + 4 \rightarrow \text{PC}$ 控制信号: 见教材P211
- 译码/取数阶段T2: $R[\text{rs}] \rightarrow \text{A}$ $R[\text{rt}] \rightarrow \text{B}$ $\text{PC} + \text{SignImm} \ll 2 \rightarrow \text{PC}$ 控制信号: 见教材P212
- 执行周期: $\text{A} \rightarrow \text{SrcA}$ $0 \rightarrow \text{SrcB}$ ALU做减法运算 如果结果 > 0 ，则 $\text{PC} + 4 + \text{SignExt}(\text{imm}) \ll 2 \rightarrow \text{PC}$ ；控制信号: $\text{ALUSrcA}=1$, $\text{ALUSrcBZero}=1$, $\text{SUB}=1$, $\text{Bgtz}=1$

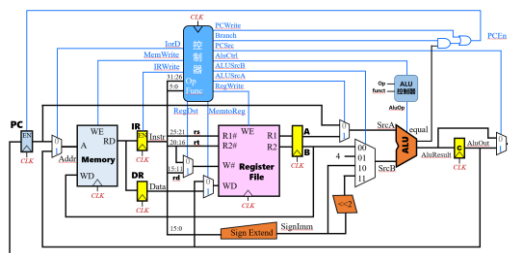


图6.27 多周期MIPS处理器的数据通路

- 6.11

- 答:

- 图6.30中的多周期数据通路R型运算指令的执行阶段需要2个时钟周期:

- T1: $A \text{ op } B \rightarrow C$; 数据运算, 结果送C寄存器, 具体执行什么运算由IR中的Op和funct字段译码决定(图中的绿色通路, 时钟周期T3内执行)。
- $C \rightarrow R[rd]$; 结果写回, 将运算结果从寄存器C, 写回到寄存器堆的rd寄存器中(图中的红色通路, 时钟周期T4内执行)。

- 如果在ALU的输出端与寄存器堆的WD端之间直接建立一个数据通路, 则可以在1个时钟周期内完成R型运算指令的执行阶段:

- T1: $A \text{ op } B \rightarrow R[rd]$

- 优化后, lwswbeqR型运算 (add)、I型运算 (addi) 指令的CPI分别为: 5、4、3、3、4。

- 题6.10的平均CPI= $5 \times 10\% + 4 \times 10\% + 3 \times 10 + 3 \times 50\% + 4 \times 20\% = 3.5$; 执行时间= $1000 \times 10^8 \times 3.5 \times 200 \text{ps} = 70 \text{s}$

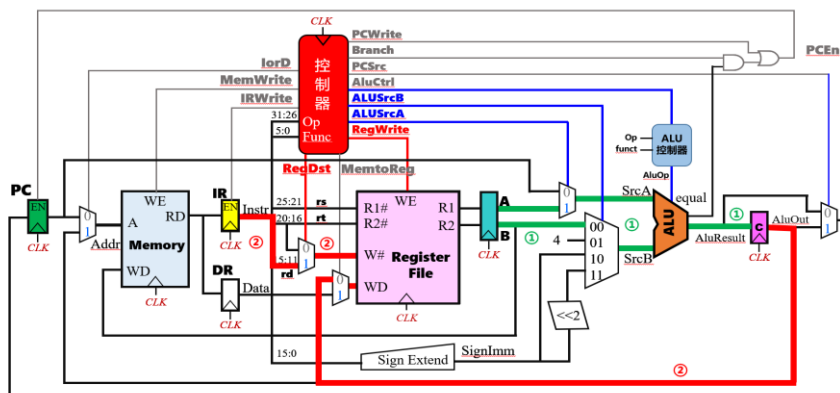


图6.30 R型算术逻辑运算指令执行阶段的数据通路

- 6.12
- 答:
- 多周期MIPS处理器的**最小时钟周期**: $T_{\min_clk} = T_{clk_to_q} + T_{mux} + \max(T_{alu} + T_{mux}, T_{mem}) + T_{setup} = 20 + 20 + \max(90 + 20, 150) + 10 = 40 + 150 + 10 = 200ps$
- 可以看到，存储器的延迟（ $T_{mem}=150$ ）是一个瓶颈，因此应该选择优化存储器的延迟
- 当存储器的延迟减少到 $T_{mem}=110$ 时，处理器性能最优，而成本最低（减少存储器延迟所花费的成本）

- 6.20

- 答：

- (1)

- 控制存储器容量=128x32，因此微地址长度=7位 ($2^7=128$)
- 判别测试需要3位，则操作控制字段=32-7-3=22位
- 微指令的3个字段分别为：22位（操作控制字段）、3位（判别测试字段）、7位（下址字段）

- (2)

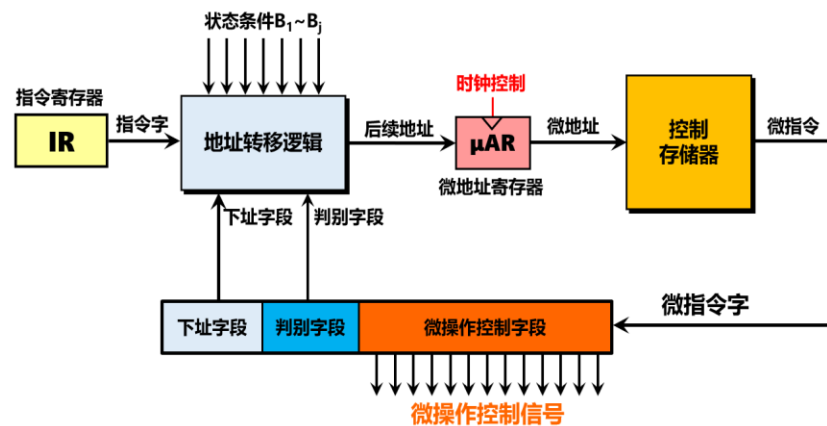


图6.49 微程序控制器组成框图（下址字段法）

- 6.21

- 答：根据题目的表6.22，可以得到下面的表：

	a	b	c	d	e	f	g	h	i	j
μl_1	√		√		√		√			
μl_2	√			√		√		√		√
μl_3	√			√	√					
μl_4	√	√							√	
μl_5	√			√		√				√

- 根据这个表可知，(b,c,d)是互斥的微操作，即b、c、d分别用于5个不同的微指令
- 同理，(e,f,i)也是互斥的微操作
- 因此(b,c,d)和(e,f,i)可以采用字段译码法进行编码，每个字段需要2位，而a、g、h、j则采用直接表示法，共需要2+2+4=8位，节省了2位（如果都采用直接表示法，则需要10位）

- 6.23
- 答:
- **ADD (R1), R0 指令的功能: $(R0) + ((R1)) \rightarrow (R1)$**
- 该指令的执行阶段的功能及控制信号如下:

时钟	功能	有效控制信号
C5	$MAR \leftarrow (R1)$	$R1_{out}, MAR_{in}$
C6	$MDR \leftarrow M(MAR)$ $A \leftarrow (R0)$	$MemR, MDR_{in}E$ $R0_{out}, A_{in}$
C7	$AC \leftarrow (MDR) + (A)$	MDR_{out}, Add, AC_{in}
C8	$MDR \leftarrow (AC)$	AC_{out}, MDR_{in}
C9	$M(MAR) \leftarrow (MDR)$	$MDR_{out}E, MemW$

Thanks