

第三章习题解答

1. 微处理器的三种基本操作分别是什么？

解答：1) 通过使用ALU（算术/逻辑单元），微处理器执行数学计算。

2) 将数据从一个内存位置移动到另一个位置。

3) 做出决定，并根据这些决定跳转到一组新指令。

2. 微处理器的基本构成包括哪些部分？这些部分分别起什么作用？

解答：

简单微处理器构成：

1. 算术逻辑运算单元 ALU，实现各种算术逻辑运算。
2. 寄存器组包括通用寄存器、指令寄存器、程序指针计数器等。通用寄存器用来暂存参与 ALU 单元运算的数据和中间结果，指令寄存器用来暂存从内存读入的指令，程序计数器用来指示下一条要执行的指令在内存中的存放地址。
3. 指令译码器实现指令的译码，并根据译码结果控制 CPU 完成相关操作。

3. 微处理器的数据通路包括哪些部件？分别完成什么功能？

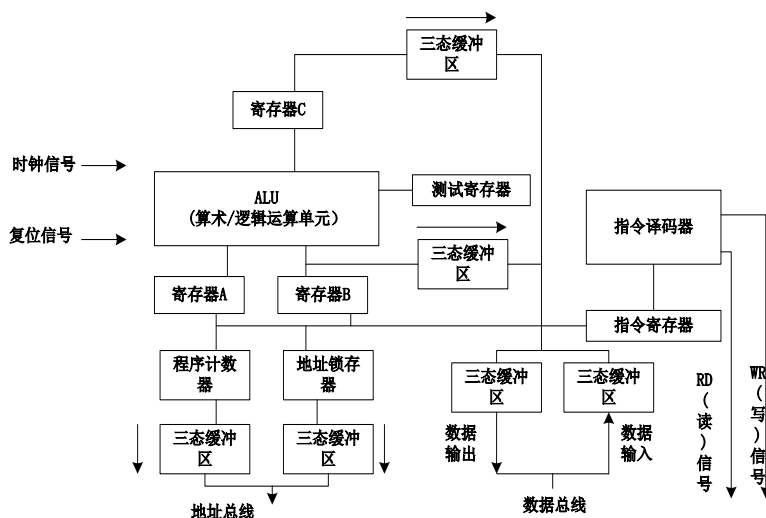
解答：

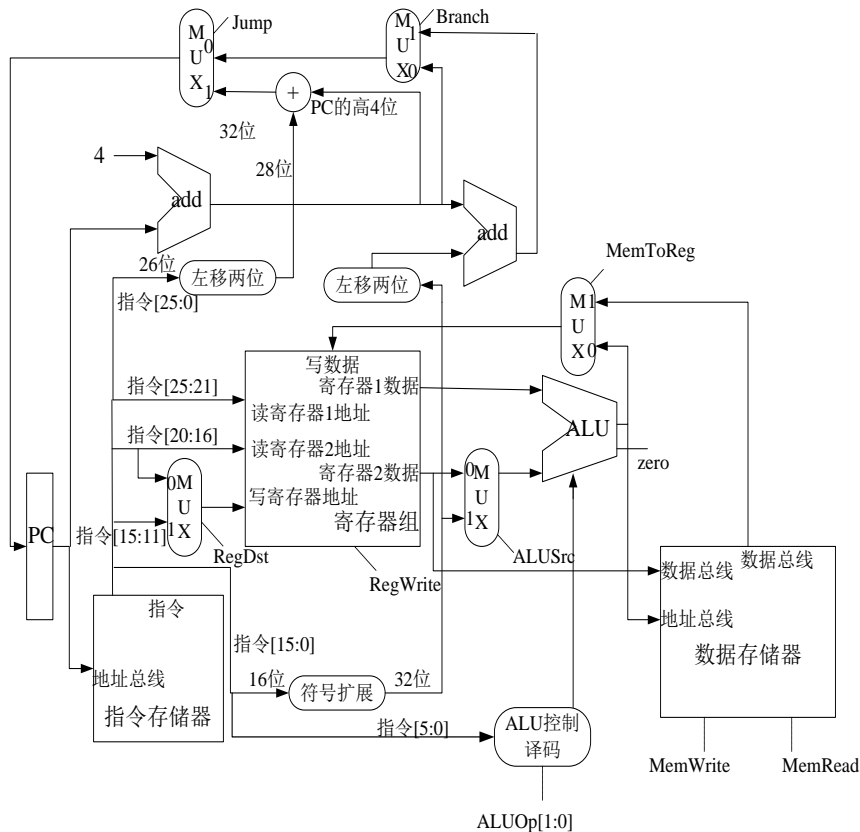
微处理器的数据通路包括：

- 1) 指令存储器，存储程序指令
- 2) 数据存储器，存储数据
- 3) 寄存器组，暂存数据和中间结果
- 4) ALU，进行算术逻辑运算
- 5) PC指针，计算下个PC的值
- 6) 符号扩展，16位立即数扩展为32位

4. 试对比分析图3- 1 与图3- 12 微处理器构成的异同。

解答：





相同处：ALU，寄存器组，指令指针，指令译码

不同处：3.1指令寄存器，CPU外部总线，无数据存储器、指令存储器；3.12 指令存储器、数据存储器，无外部总线；其不同点存在的原因：3.1为非嵌入式CPU的结构，3.12为嵌入式CPU的结构，且默认不扩展任何外设以及存储器。

5. 举例分别简述MIPS 三类指令的执行过程。

解答：1) 运算类指令直接寄存器组或立即数扩展后提供数据给ALU单元进行运算后，将结果保存到寄存器组中。

2) 数据存取类指令首先由寄存器组以及立即数扩展提供数据给ALU单元计算获得数据存储器地址，然后再根据存、取的要求将数据从寄存器组搬到数据存储器或反之。

3) 程序控制类指令根据ALU部件运算的结果控制PC部件计算下一条指令的地址。

6. 采用Verilog HDL 语言实现一个能执行以下指令集的简单微处理器：

a) lw, sw指令

b) add, sub, and, or, slt指令

c) beq, j 指令

解答：参见《计算机组成原理与接口技术——基于MIPS架构实验教程》

7. 在题6 指令集基础上，增加addu, addiu指令，修改图3- 9 简单MIPS 微处理器数据通路构成，并完成ALU 控制器和主控制器译码设计。

解答：addu, addiu指令进行无符号加运算，若在题6没有考虑溢出标志，那么ADD与ADDU指令的执行效果是一样的，因此可以不做修改。但是ADDIU由于是立即数参与运算，题6没有考虑，因此需要重新考虑该指令。

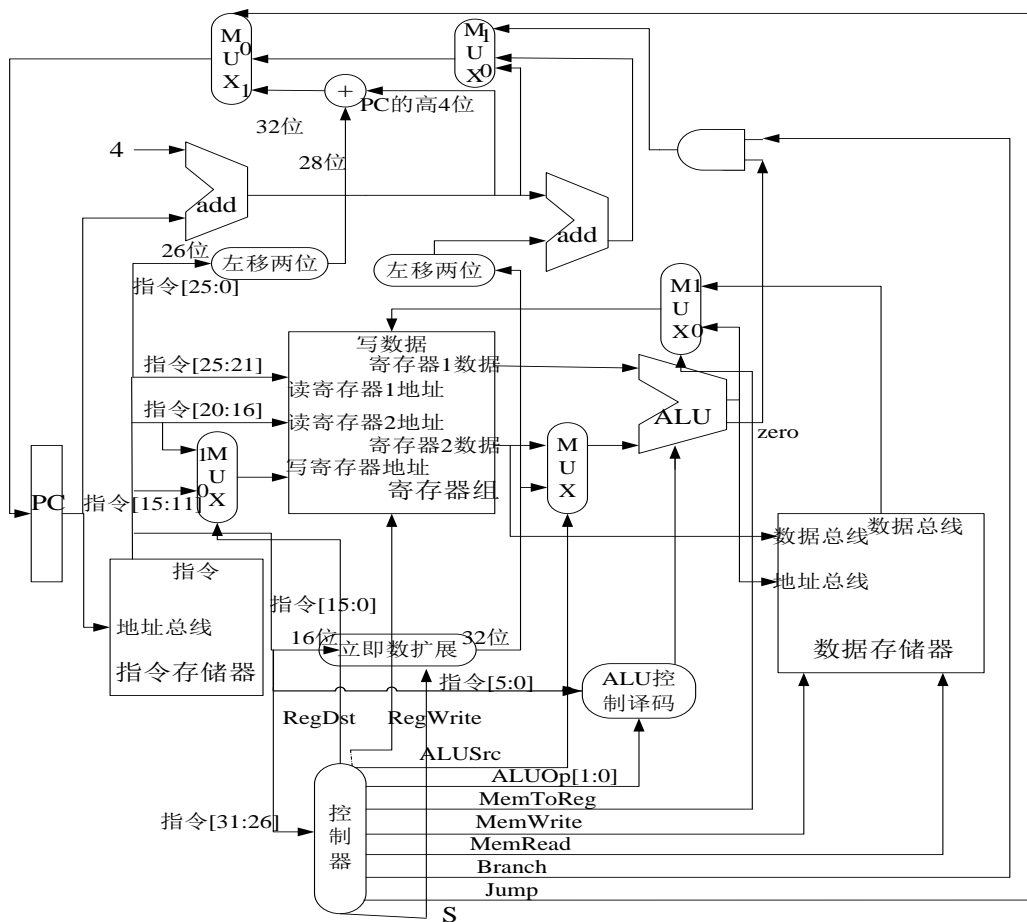
ADDIU指令为I型指令，且执行加法运算，因此对ALU控制信号的编码方式与LW，SW指令相同，但是主控制器的控制信号编码有所区别，根据下表个控制信号的定义

控制信号名称	置 1	清 0
RegDst	表示写寄存器的地址来自指令[15:11]	来自指令[20:16]
Jump	表示 PC 的值来自伪直接寻址	来自另一个复用器
Branch	表示下一级复用器的输入来 PC 相对寻址加法器	来自 PC+4
MemToReg	表示写寄存器数据来自存储器数据总线	来自 ALU 结果
ALUSrc	表示 ALU 的第二个数据源来自指令[15:0]	来自读寄存器 2
RegWrite	将写寄存器数据存入写寄存器地址中	无操作
MemWrite	将写数据总线上的数据写入内存地址单元	无操作
MemRead	将内存单元的内容输出到读数据总线上	无操作

可以得到各位的值为：

指令	Reg Dst	Jum p	Bran ch	MemToRe g	ALUSr c	RegWrit e	MemWrit e	MemRea d	ALUOp 1	ALUOp 0	S
addiu	0	0	0	0	1	1	0	0	0	0	0
R 型	1	0	0	0	0	1	0	0	1	0	X
lw	0	0	0	1	1	1	0	1	0	0	1
sw	X	0	0	X	1	0	1	0	0	0	1
beq	X	0	1	X	0	0	0	0	0	1	1
j	X	1	0	X	X	0	0	0	X	X	X

但是由于该指令为无符号数运算，因此数据通路上立即数的扩展部分需要增加无符号扩展，给立即数扩展增加一个控制信号S，其余I型指令S=1，表示符号扩展，ADDIU指令S=0，表示无符号扩展。此部分可通过对操作码进行译码获得。因此主控制器增加一位译码输出。整个微处理器结构如下图所示。



8. 简述流水线技术与超标量技术的特点。

解答：

在同一条指令中顺序执行，在不同的指令中并行执行，这就是流水线的基本原理。

微处理器集成多个 ALU、多个译码器和多条流水线，以并行处理的方式来提高性能，这就是超标量技术。通过采用超标量技术可以提高指令级并行度。

流水线处理器在每个时钟周期最多发出一条指令，而超标量处理器可以在每个时钟周期发出 1~8 条指令。

9. 微处理器识别异常事件的方法有哪些？它们各有什么优缺点？

解答：

第一种方式是状态位法：在微处理器中利用一个寄存器对每种异常事件确定一个标志位，当有异常事件发生时，寄存器中对应的位置 1，一个 32 位的寄存器可以表示 32 种不同类型的异常事件。第二种方法是向量法：对不同类型的异常事件进行编码，这个编码叫中断类型码或异常类型码。若产生了异常事件，就产生相对应的异常类型码，微处理器通过解码就可以获知异常发生的原因。如果采用一个 8 位的寄存器记录异常类型码，就可以实现 256 种不同类型的异常事件识别。

优缺点：状态位法识别的中断类型有限，但是硬件设计简单，不需要设计编码器；向量法识别的中断类型较多，但是需要专门的中断编码器，硬件设计复杂一些。

10. 微处理器保存断点的方法有哪些？它们各有什么优缺点？

解答：

第一种方式是在微处理器中设计一个寄存器EPC，当微处理器出现异常时，就将PC的值保存到EPC中。异常处理完之后，再把EPC的值赋给PC，这样就可以实现中断的返回。但是采用这种方式，如果计算机正在处理中断还没有返回，就不能处理再次发生的异常事件，就如同仅采用\$ra保存主程序的返回地址类似。因此为实现异常处理的嵌套，微处理器需要在进入异常处理程序之前首先将EPC的值压入栈中，然后再保存PC的值到EPC中。

第二种方式是当出现异常时，微处理器直接将PC的值压入栈中，异常处理完之后，再从栈顶把值弹出来赋给PC，这样也就不用担心异常处理的嵌套问题了。

优缺点：第一种方式的优点是中断返回速度快，无需访问内存，缺点是中断嵌套需要用户中断服务程序软件实现压栈和出栈的管理；第二种方式优点是可以任意次数的中断嵌套，无需用户程序管理，缺点是每次中断进入和返回CPU都将访问内存，因此速度稍慢。

11. 微处理器进入异常处理的方法有哪些？它们各有什么优缺点？

解答：

1) 微处理器分配一块专门的内存区域保存异常处理程序。通常在这块内存区域中为每个异常处理程序分配固定长度的空间如 32 个字节或 8 条指令长度的空间，而且针对每个异常事件其异常处理程序的存放地址是固定的。如果 8 条指令不能完成异常处理，那么在这 32 个字节的内存空间中就只能实现一些简单的处理，然后再通过一条跳转指令跳转到真正的异常处理程序的存放地址处。这种方式适合于大多数异常处理非常简单的应用场景。

2) 微处理器为所有的异常事件仅提供一个异常处理程序存放地址，发生任何异常事件都首先转移到该地址执行总的异常处理，并在总异常处理程序中分析异常事件的原因，然后再根据异常的原因通过子程序调用的方式去执行相应的异常处理。在这种实现方式中，微处理器需要逐个比较异常状态标志寄存器识别异常事件。硬件实现简单，但是降低了异常处理的效率。

3) 微处理器分配一块专门的内存区域保存异常处理程序的入口地址，这个入口地址也叫中断向量。保存异常处理程序的入口地址的内存区域叫做中断向量表。每个中断向量都是固定长度的，如 4 个字节等。针对每个异常事件其异常处理程序的入口地址存放位置是固定的。在这种方式中，异常处理程序可以存放在内存中的任意位置，只需要把该异常处理程序的入口地址保存到中断向量表中正确的地址中，当异常发生时，微处理器就可以通过中断向量表查找到中断服务程序的入口地址。这种方式是间接获取异常处理程序的地址。

1) 2) 的部分工作需要软件来完成，硬件设计简化，适合于嵌入式系统；软件稍复杂，且进入中断服务较慢；3) 纯硬件的实现方式，硬件设计复杂，但是进入中断快。

12. 参考 MicroBlaze 数据手册，说明 MSR 寄存器各位的具体含义。

解答：

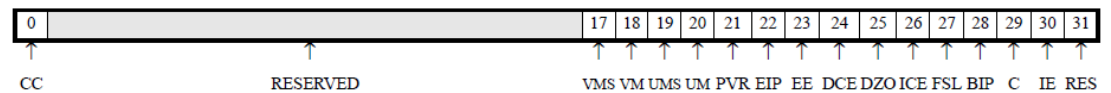


Figure 1-4: MSR

VMS: 虚拟保护模式保存

VM: 虚拟保护模式

UMS: 用户模式保存

UM: 用户模式

PVR: 处理器版本寄存器

EIP: 异常处理中

EE: 异常使能
DCE:数据 cache 使能
DZO: 除法溢出
ICE: 指令 cache 使能
FSL: FSL 总线出错
BIP: 打断处理中
C: 进位标识
IE: 中断使能
RES: 保留
CC: 进位拷贝与 C 相同