

32位MIPS处理器实验需求文档

SHS小组

乔奕 郭嘉丞 陈雨兰

目 录

1 引言

撰写本需求文档的目的在于，明确项目目标和具体需要完成的功能，其中具体讨论了CPU 的模块设计和与操作系统的相关接口，对于操作系统中与CPU 实现相关不大的部分并不涉及，如有需要，可阅读本小组的操作系统移植实验记录。

我们实现的CPU 为一个无缓存的多时钟流水线CPU，操作系统为清华大学教学操作系统ucore。

1.1 背景

系统名称：32位MIPS处理器

任务提出者：刘卫东老师 白晓颖老师

开发者：计32 乔奕 计44 郭嘉丞 计35 陈雨兰

1.2 实验目标

1. 使用提供的开发板,在FPGA 上编程实现一个基于标准32 位MIPS 指令集的子集的流水CPU,支持异常、中断、TLB 等。
2. 在该CPU 上运行ucore 操作系统,进入用户态及shell 环境,正常执行shell 命令。
3. 修改ucore, 实现简单的远程文件执行功能,即通过串口从PC 上获取ELF 文件,并在本地执行。
4. 可选实现对VGA、ps/2 keyboard 等其它外设的支持。

1.3 参考资料

《实验指导文档》

《计算机组成与设计 硬件/软件接口》

2 功能需求

2.1 ALU

ALU 负责双操作数的算术逻辑运算，由ALUOp 信号和指令的funct 字段控制，完成数据和地址的算术、逻辑、移位运算，输出结果。其中，乘法指令不通过ALU 计算，由乘法器支持，PC + 4 运算由专门的加法器支持。

为支持流水线CPU 的正常运行，部分运算如跳转指令的地址计算不通过ALU。

2.2 乘法器

乘法器的实现独立于ALU，最后的乘法结果为64位二进制，写入HI和LO寄存器。
Xilinx FPGA的IP核可支持组合逻辑乘法。

2.3 寄存器堆

对于32位MIPS系统，需要在寄存器堆中实现32个通用32位寄存器，在指令解码阶段读取寄存器内容，在写回阶段完成写入。

2.4 CP0

系统控制协处理器CP0主要提供管理CPU资源所需的机制，包括MMU、TLB与异常处理控制。通过调用MFC0，MTC0指令，CP0提供了统一的对外接口以完成对寄存器组的访问。

下表为需要实现的CP0寄存器及其主要功能。

编号	寄存器名称	寄存器功能
0	Index	用于TLBWI指令访问TLB入口的索引序号
2	EntryLo0	作为TLBWI及其他TLB指令接口，管理偶数页入口
3	EntryLo1	作为TLBWI及其他TLB指令接口，管理奇数页入口
9	BadVAddr	捕捉最近一次地址错误或TLB异常时的虚拟地址
10	Count	每隔一个时钟增加1，用作计时器，并可使能控制
11	EntryHi	TLB异常时，系统将虚拟地址部分写入EntryHi寄存器中用于TLB匹配信息
12	Compare	Compare保持一定值，当Count值与Compare相等时，SI_TimerInt引脚变高电平直到有数值写入Compare，用于定时中断
13	Status	表示处理器的操作模式、中断使能及诊断状态
15	Cause	记录最近一次异常原因，控制软件中断请求和中断处理派分向量
16	EPC	存储异常处理之后程序恢复执行的地址
18	EBase	识别多处理器系统中不同的处理器异常向量的基地址

2.4.1 Index

Index寄存器是一个32位的读/写寄存器，可用于TLBP、TLBR和TLBWI指令访问TLB入口的索引序号。

比特位	描述	读/写	重置状态
31	检测故障，TLPB指令没有在TLB中寻得匹配时置1	R/W	未定义
30:6	必须写为0，读取返回0	0	0
5:0	TLB入口索引，受TLBRead和TLBWrite影响	R/W	未定义

2.4.2 EntryLo1/EntryLo0

EntryLo 寄存器的作用等同于TLB、TLBR、TLBWI 和TLBWR 指令时的接口。选基于TLB 的MMU 而言，EntryLo0 管理偶数页的入口，EntryLo1 管理奇数页的入口。如果出现地址错误，TLB 失效，TLB 修改或是TLB 重填异常的行为，寄存器的内容将成为未定义的。

比特位	描述	读/写	重置状态
31:30	保留区域，写入忽略，读取返回0	R	0
29:26	强制写入0	R	0
25:6	页帧号PFN，对应了物理地址31:12位	R/W	未定义
5:3	页面一致性属性C	R/W	未定义
2	写使能位D，D=1则允许写入该页	R/W	未定义
1	有效位V，V=1则允许访问该页，V=0引起TLB无效异常	R/W	未定义
0	全局位G，两个寄存器的G位共同决定TLB匹配中是否忽略ASID比较	R/W	未定义

2.4.3 EntryHi

EntryHi 寄存器包含了用于TLB 读、写和访问操作的虚拟地址匹配信息。当TLB 异常发生时，系统将虚拟地址的[31:13]位写入EntryHi 寄存器的VPN2 区域。TLBR 指令将选中的TLB 入口相应的区域写入EntryHi 寄存器。操作系统将当前地址空时标识符写入ASID 区域，该区域在TLB 比较过程中用于确定TLB 是否可以匹配。

比特位	描述	读/写	重置状态
31:13	VPN2，TLB异常时由硬件写入	R/W	未定义
12:8	必须写入0，返回0	0	0
7:0	地址空间标识符ASID，在TLB读取时由硬件写入	R/W	未定义

2.4.4 Status

Status 寄存器的区域联合作用，可以创建协处理器的工作模式。当以下所有条件都成立时启用中断：

Status[0]:IE = 1

Status[0]:EXL = 0

Status[0]:ERL = 0

此时设置IM（Status[16:9]）位和IE 位可以使能中断。

EXL 与ERL 任一位置1 都可使系统进入Kernel 模式，否则为User 模式。异常处理开始时，将Status[1] 赋值为1，在执行ERET 指令时将Status[1]赋值为0。

2.4.5 Cause

Cause 寄存器记录了最近一次异常的原因，也控制软件中断请求以及中断处理派分的向量。Cause[6:2] 表示异常号。

2.4.6 Count/Compare

每经过一个周期，Count 自增1，需内置一个加法器。

详情见异常处理的时钟中断部分。

2.4.7 EPC

如果Cause(BD)为1，也就是异常发生在延迟槽中，实际发生异常的指令就为EPC+4。但是我们返回执行的位置仍然应该是EPC，否则分枝指令的跳转就会无法执行。因此CPU必须在内部记录指令是否在延迟槽内。如果在延迟槽内的指令发生了异常，应该设置PC=EPC-4。

2.4.8 Ebase

EBase 寄存器是一个读/写寄存器，包含了在StatusBEV 为0 时所使用的异常向量的基地址及一个只读CPU 号，该CPU 号可以被软件用来区分多处理器系统中不同的处理器。

比特位	描述	读/写	重置状
31:30	固定为10，从而强制异常基地址在kseg0 或kseg1 的无映射的虚拟地址段中	R	10
29:12	在StatusBEV 为0 时指定异常的基地址	R/W	0
9:0	CPUNum	R	外部设
11:10	必须写入0，返回0	0	0

2.5 MMU

内存管理单元MMU 通过TLB异常实现。我们需要在CPU上实现一个TLB列表，每次访存需要通过TLB列表把线性地址（即虚拟地址）转化为物理地址。此外，还需要实现TLB重填的功能，包括设计若干CP0寄存器，并实现TLB重填指令tlbwi。

MIPS 内存地址分配 本实验中，地址0x80000000以上，属于kseg0区域，供操作系统内核使用，这一区域不进行map，因此，在我们实现的MMU 中，应当判断地址是否大于0x80000000，若是，则不进行映射，若否，进入TLB查找阶段。

从0x00000000到0x80000000是KUSEG区域，这块内存会被用于用户进程的虚存分配。

通过TLB异常实现MMU CPU发起访存，使用了32位线性地址。TLB模块抽取线性地址中的前20位，作为VPN，在TLB表中查找。若查找得到，则根据表中的PPN，结合offset得到物理地址，然后直接访存。

若无法找到，将该线性地址传入CP0中的vaddr寄存器，然后触发TLBmiss异常，进入异常处理程序。操作系统取出vaddr中的值，即访存失败的线性地址。若地址所在页表不存在，则建立新表。之后，把新表所对应的物理地址，存入CP0寄存器中。之后，操作系统利用汇编语句直接调用tlbwi。此时，CPU根据实现了的tlbwi指令，利用CP0寄存器中的值，重新填充TLB中的某一项。此处，轮流重填TLB中的项（也可以随机填充）。

MIPS异常机制会重新执行上一条代码，此时能够正常访存。

2.6 异常处理

本实验中，由硬件检测到异常发生，填写相应的CP0寄存器后，跳转到操作系统的异常处理函数。操作系统根据CP0寄存器的值判断异常的类型，分发到相应的处理代码。

2.6.1 异常向量

MIPS32 CPU上有两组异常处理向量，根据SR(BEV)位(SR寄存器的bit22)切换：

BEV==1: ROM上的异常处理 这是我们CPU刚刚启动时处于的状态。由于操作系统的启动还没有完成，我们的异常处理向量被放在ROM固件中。由于没有缓存系统，所以没有缓存错误的入口点。我们的CPU也没有支持EIC异常。由于我们的操作系统还没有启动，我们在实现中简单的将所有的ROM异常都直接进行无限循环。我们将会使用具体的异常向量表如下：

入口地址	类型
0xBFC00200	简单的TLB重填
0xBFC00380	其他所有异常

BEV==0: RAM上的异常处理 当操作系统完成自己的异常向量的装载时，就会将此位置零。并且向C0寄存器堆中的EBase寄存器写入我们的异常向量的基址。实际上的异常向量基址Base为EBase[29:12]&0...0，此时我们会用到的异常向量表如下：

入口地址	类型
BASE+0x000	简单的TLB重填
BASE+0x180	其他所有异常

2.6.2 精确异常处理流程

1. 保存中断信息，主要是EPC，BadVAddr，Status，Cause 等寄存器的信息。

EPC: 存储异常处理之后程序恢复执行的地址。对于一般异常，当前发生错误的指令地址即为EPC 应当保存的地址；而对于硬件中断，由于是异步产生，则可以任意设定一条并未执行完成的指令地址保存，但在进入下一步处理之前，该指令前的指令都应当被执行完。

BadVAddr: 捕捉最近一次地址错误或TLB 异常（重填、失效、修改）时的虚拟地址。

Status: 将EXL 位置为1，进入kernel 模式进行中断处理

Cause: 记下异常号。

EnrtyHi: tlb 异常时，记下BadVAddr 的部分高位。

2. 根据Cause 中的异常号跳转到相应的异常处理函数入口
3. 中断处理
4. 通过调用ERET 指令恢复现场，返回EPC 所存地址执行并且将Status 中的EXL 重置为0 表示进入user 模式。

2.6.3 异常描述

异常与中断列表如下：

异常号	异常名	描述
0	Interrupt	外部中断，异步发生，由硬件引起
1	TLB Modified	内存修改异常，发生在Memory 阶段
2	TLBL	读未在TLB 中映射的内存地址触发的异常
3	TLBS	写未在TLB 中映射的内存地址触发的异常
4	ADEL	读访问一个非选节地址触发的异常
5	ADES	写访问一个非选节地址触发的异常
8	SYSCALL	系统调用
10	RI	执行未定义指令异常

可能用到的中断号如下：

中断号	描述
4	串口中断
6	键盘中断
7	时钟中断

2.6.4 硬件中断

时钟中断 由操作系统读取count寄存器并设置compare寄存器的值，Count寄存器自增到和Compare相等引起中断。此时填写cause寄存器的中断号，由操作系统处理中断。

外设中断 与时钟中断类似，但是设置不同的硬件中断号来表示中断类别。可能涉及的外设有串口和键盘。

2.7 串口

我们预计实现如下串口功能：

1. 串口的数据地址映射到0xbfd003f8上，状态地址映射到0xbfd003fb
2. 状态地址提供是否有新输入数据的标志位bit0，其他标志位保留
3. 提供中断功能，当串口有新数据时引起串口异常。

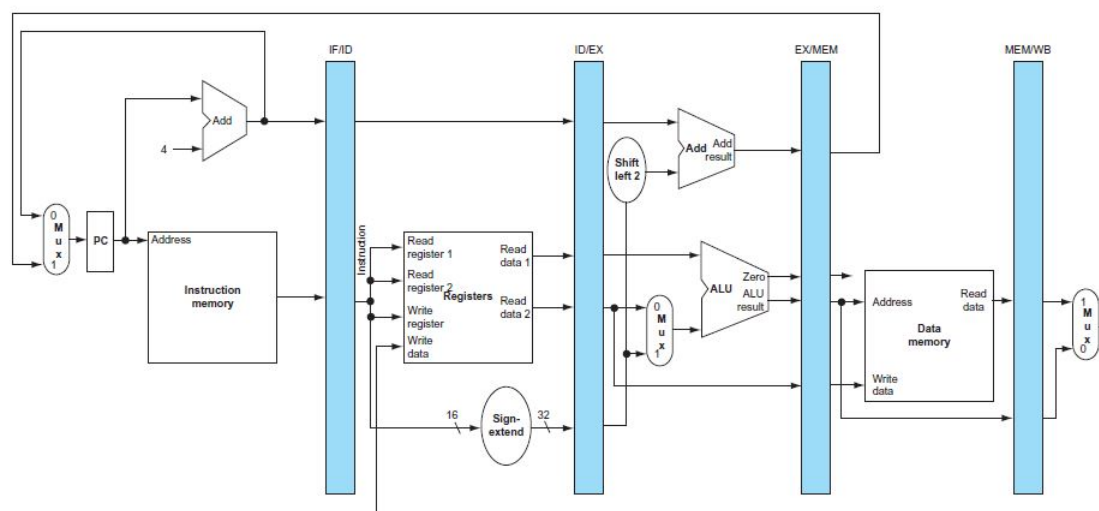
2.8 指令集

我们采用的是MIPS32 的标准子集作为指令集，共计48 条指令，每一条指令是一个32 位字。由于编译器版本不同，无法保证囊括所有所需指令，可能需要实现这48 条以外的指令。

具体的指令内容详见附录A。

2.9 数据通路

流水线的数据通路示意图如下。（出自《计算机组成与设计 硬件/软件接口》4.6 节）



3 性能需求

实现多周期流水CPU，用旁路处理数据冒险。主频取决于流水线耗时最长的模块。

4 运行环境

主要硬件设备信息如下：

FPGA	Xilinx Spartan6 xc6slx100
RAM	32-bit 字长，4 块，共8MB
Flash	16-bit 字长，共8MB
CPLD	与FPGA 相连，用于I/O
以太网接口	100MB

A MIPS指令集

表中rs, rt, rd 为寄存器编号, immediate 为立即数。

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	0	0	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	ADDIU rt is immediate															
指令功能	$R[t] \leftarrow R[s] + \text{Sign-extend}(\text{immediate})$															
功能说明	对立即数进行符号扩展后与寄存器rs的值求和, 结果保存到寄存器rt中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	0	0	1
MIPS语言	ADDU rd rs rt															
指令功能	$R[d] \leftarrow R[s] + R[t]$															
功能说明	将寄存器rs与寄存器rt的值求和, 结果保存到寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	1	0	1	0
MIPS语言	SLT rd rs rt															
指令功能	if($R[s] < R[t]$) then $R[d] = 1$, else $R[d] = 0$															
功能说明	比较寄存器rs与寄存器rt的值并根据结果对寄存器rd赋值															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	0	1	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	SLTI rt rs immediate															
指令功能	if($R[s] < \text{Sign-extend}(\text{immediate})$) $R[t] = 1$, else $R[t] = 0$															
功能说明	比较寄存器rs与立即数进行符号扩展后的值并根据结果对寄存器rt赋值															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	0	1	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	SLTIU rt rs immediate															
指令功能	if($R[s] < \text{Zero-extend}(\text{immediate})$) $R[t] = 1$, else $R[t] = 0$															
功能说明	比较寄存器rs与立即数进行零扩展后的值并根据结果对寄存器rt赋值															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	1	0	1	1
MIPS语言	SLTU rd rs rt															
指令功能	if($R[s] < R[t]$) $R[d] = 1$, else $R[d] = 0$															
功能说明	比较寄存器rs与寄存器rt的值并根据结果对寄存器rd赋值															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	0	1	1
MIPS语言	SUBU rd rs rt															
指令功能	$R[d] \leftarrow R[s] - R[t]$															
功能说明	用寄存器rs的值减寄存器rt的值，结果保存到寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
MIPS语言	MULT rs rt															
指令功能	$HI/LO \leftarrow R[s] * R[t]$															
功能说明	将寄存器rs与寄存器rt的值相乘，保存到寄存器HI/LO中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	1	0	0	1	0
MIPS语言	MFLO rd															
指令功能	$R[d] \leftarrow LO$															
功能说明	将LO寄存器的值保存到rd寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	1	0	0	0	0
MIPS语言	MFHI rd															
指令功能	$R[d] \leftarrow HI$															
功能说明	将HI寄存器的值保存到rd寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
MIPS语言	MTLO rs															
指令功能	$LO \leftarrow R[s]$															
功能说明	将寄存器rs的值保存到LO寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
MIPS语言	MTHI rs															
指令功能	$HI \leftarrow R[s]$															
功能说明	将寄存器rs的值保存到HI寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	1	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BEQ rs rt immediate															
指令功能	if($R[s] = R[t]$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs与寄存器rt的值相等，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	1	rs					0	0	0	0	1
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BGEZ rs immediate															
指令功能	if($R[s] \geq 0$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs的值大于等于0，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	1	1	1	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BGTZ rs immediate															
指令功能	if($R[s] > 0$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs的值大于0，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	1	1	0	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BLEZ rs															
指令功能	if($R[s] \leq 0$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs的值小于等于0，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	1	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BLTZ rs															
指令功能	if($R[s] < 0$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs的值小于0，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	1	0	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	BNE rs rt															
指令功能	if($R[s] \neq R[t]$) $PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	如果寄存器rs与寄存器rt的值不相等，则跳转到目的地址执行，否则顺序执行下一条指令															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	1	0	immediate(26bit)									
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate(26bit)															
MIPS语言	J immediate															
指令功能	$PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$															
功能说明	无条件跳转目的地址执行															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	1	1	immediate(26bit)									
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate(26bit)															
MIPS语言	JAL immediate															
指令功能	$PC \leftarrow PC + \text{Sign-extend}(\text{immediate})$, $RA \leftarrow PC$															
功能说明	无条件跳转目的地址执行，将延迟槽后一条指令存入RA															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	1	0	0	1
MIPS语言	JALR rs rd															
指令功能	$PC \leftarrow R[s], R[d] \leftarrow RPC$															
功能说明	无条件跳转目的寄存器rs中所存地址执行，将延时槽后一条指令存入R[d]															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
MIPS语言	JR rs															
指令功能	$PC \leftarrow R[s]$															
功能说明	无条件跳转至寄存器rs所存地址执行															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	0	0	1	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	LW rt rs immediate															
指令功能	$R[t] \leftarrow MEM[R[s] + \text{Sign-extend}(\text{immediate})]$															
功能说明	将寄存器rs的值与立即数immediate符号扩展后相加所得存至rt中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	1	0	1	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	SW rt rs immediate															
指令功能	$MEM[R[s] + \text{Sign-extend}(\text{immediate})] \leftarrow R[t]$															
功能说明	将寄存器rt的值存入寄存器rs的值与立即数immediate符号扩展后相加所得地址中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	LB rt rs immediate															
指令功能	$R[t] \leftarrow \text{Sign-extend}(\text{MEM_Byte}[R[s] + \text{Sign-extend}(\text{immediate})])$															
功能说明	将寄存器rs的值与立即数immediate符号扩展后相加所得地址中第一个字节取出来符号扩展后保存在寄存器rt中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	0	1	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	LBU rt rs immediate															
指令功能	$R[t] \leftarrow \text{Zero-extend}(\text{MEM_Byte}[R[s] + \text{Sign-extend}(\text{immediate})])$															
功能说明	将寄存器rs的值与立即数immediate符号扩展后相加所得地址中的第一个字节取出来零扩展后保存在寄存器rt中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	1	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	SB rt rs immediate															
指令功能	$\text{MEM_Byte}[R[s] + \text{Sign-extend}(\text{immediate})] \leftarrow \text{LOW_BYTE}[R[t]]$															
功能说明	将寄存器rt的最低字节取出来保存在rs的值与立即数immediate符号扩展后相加所得地址中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	1	0	0
MIPS语言	AND rd rs rt															
指令功能	$R[d] \leftarrow R[s] \& R[t]$															
功能说明	将寄存器rs与寄存器rt的值相与后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	1	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	ANDI rt rs immediate															
指令功能	$R[t] \leftarrow R[s] \& \text{Zero-extend(immediate)}$															
功能说明	将寄存器rs的值与立即数零扩展后相与的结果保存至寄存器rt中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	1	1	1	0	0	0	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	LUI rt immediate															
指令功能	$R[t] \leftarrow \text{immediate} * 65536$															
功能说明	将16为立即数放至寄存器rt的高16位中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	1	1	1
MIPS语言	NOR rd rs rt															
指令功能	$R[d] \leftarrow \sim(R[s] R[t])$															
功能说明	将寄存器rs与寄存器rt的值或非后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	1	0	1
MIPS语言	OR rd rs rt															
指令功能	$R[d] \leftarrow R[s] R[t]$															
功能说明	将寄存器rs与寄存器rt的值相或后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	1	0	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	ORI rt rs immediate															
指令功能	$R[t] \leftarrow R[s] \mid \text{Zero-extend}(\text{immediate})$															
功能说明	将寄存器rs与立即数immediate零扩展后相或的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	1	0	0	1	1	0
MIPS语言	XOR rd rs rt															
指令功能	$R[d] \leftarrow R[s] \wedge R[t]$															
功能说明	将寄存器rs与寄存器rt的值异或后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	1	1	1	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	XORI rt rs immediate															
指令功能	$R[t] \leftarrow R[s] \wedge \text{Zero-extend}(\text{immediate})$															
功能说明	将寄存器rs与立即数immediate零扩展后相异或的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					immediate					0	0	0	0	0	0
MIPS语言	SLL rd rt immediate															
指令功能	$R[d] \leftarrow R[t] \ll \text{immediate}$															
功能说明	将寄存器rt中的值逻辑左移寄存器rs中的值位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	0	1	0	0
MIPS语言	SLLV rd rt rs															
指令功能	$R[d] \leftarrow R[t] \ll R[s]$															
功能说明	将寄存器rt中的值逻辑左移寄存器rs中的值位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					immediate						0	0	0	0	1
MIPS语言	SRA rd rt immediate															
指令功能	$R[d] \leftarrow R[t] \gg \text{immediate}(\text{arithmetic})$															
功能说明	将寄存器rt中的值算数右移立即数immediate位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	0	1	1	1
MIPS语言	SRAV rd rt rs															
指令功能	$R[d] \leftarrow R[t] \gg R[s](\text{arithmetic})$															
功能说明	将寄存器rt中的值算数右移寄存器rs中的值位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					immediate					0	0	0	0	1	0
MIPS语言	SRL rd rt immediate															
指令功能	$R[d] \leftarrow R[t] \gg \text{immediate}(\text{logical})$															
功能说明	将寄存器rt中的值逻辑右移立即数immediate位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	0	1	1	0
MIPS语言	SRLV rd rt rs															
指令功能	$R[d] \leftarrow R[t] \gg R[s](\text{logical})$															
功能说明	将寄存器rt中的值逻辑右移寄存器rs中的值位后的结果保存至寄存器rd中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0
MIPS语言	SYSCALL															
指令功能	中断号 \leftarrow SYSCALL															
功能说明	执行后除法中断															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	1	1	1	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	CACHE															
指令功能																
功能说明	不做cache，视为NOP															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
MIPS语言	ERET															
指令功能	$PC \leftarrow EPC$															
功能说明	返回至EPC寄存器的地址执行，并设置Status寄存器的EXL位为0															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	1	0	0	0	0	0	0	0	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	0	0	0	0
MIPS语言	MFC0 rt rd															
指令功能	$R[t] \leftarrow CP0[R[d]]$															
功能说明	将协处理器0中的rd寄存器的值保存到rt寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	1	0	0	0	0	0	0	1	0	0	rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	rd					0	0	0	0	0	0	0	0	0	0	0
MIPS语言	MTC0 rd rt															
指令功能	$CP0[R[d]] \leftarrow R[t]$															
功能说明	将寄存器rt的值保存到协处理器0中的rd寄存器中															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
MIPS语言	TLBWI															
指令功能	$R[d] \leftarrow R[t] \ll R[s]$															
功能说明	写索引TLB项															

二进制（高位）	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	0	0	1	0	1	rs					rt				
二进制（低位）	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	immediate															
MIPS语言	LHU rt rs immediate															
指令功能	$R[t] \leftarrow \text{Zero-extend}(\text{MEM_HALFWORD}[R[s] + \text{Sign-extend}(\text{immediate})])$															
功能说明	将寄存器rt中的值与立即数immediate符号扩展后相加所得地址中的低两个字节取出来零扩展后保存在寄存器rt中															