

# 32位MIPS综合实验需求文档

原子小组

October 19, 2015

## 目 录

<b>1 引言</b>	<b>2</b>
1.1 编写目的	2
1.2 背景	2
1.3 定义	2
1.4 Mips架构下CPU运行概述	2
1.5 参考资料	2
<b>2 功能需求</b>	<b>2</b>
2.1 CPU	2
2.1.1 ALU	2
2.1.2 乘法器	2
2.1.3 寄存器堆	2
2.1.4 CP0	2
2.1.5 异常中断处理	2
2.1.6 MMU	2
2.1.6.1 虚拟地址映射	2
2.1.6.2 TLB	2
2.2 Ucore	3
2.2.1 BIOS	3
2.2.2 远程文件执行	3
2.3 外设	3
2.3.1 串口	3
2.3.2 VGA	3
2.3.3 ps/2键盘	3
2.3.4 网口	3
2.4 Decaf编译器	4
2.4.1 汇编指令的生成	4
2.4.2 库函数调用及calling convention	4
2.4.3 程序入口及退出	4
2.5 指令集与数据通路	4
<b>3 性能需求</b>	<b>4</b>
<b>4 运行环境需求</b>	<b>4</b>
4.1 设备	4
4.2 控制	4
<b>5 附录</b>	<b>4</b>
5.1 指令系统	4

# 1 引言

## 1.1 编写目的

## 1.2 背景

## 1.3 定义

## 1.4 Mips架构下CPU运行概述

## 1.5 参考资料

# 2 功能需求

## 2.1 CPU

### 2.1.1 ALU

ALU负责实现双输入的算术、逻辑和移位运算功能，其中比较运算通过补码减法实现。输入为两个32位整数和一个4位符号位，输出为32位整数，不给出标志位。

ALU需要实现的运算详见下表。

操作码	功能	描述	操作码	功能	描述
ADD	$A + B$	加法	NOR	$\sim(A \mid B)$	或非
SUB	$A - B$	减法	SLL	$A \gg B$	逻辑左移
AND	$A \& B$	与	SRL	$A \ll B$	逻辑右移
OR	$A \mid B$	或	SRA	$A \ggg B$	算术右移
XOR	$A \wedge B$	异或	SLT	$A < B$	比较

### 2.1.2 乘法器

乘法器是一个独立于ALU的元件，其中乘法运算直接使用Verilog语言提供的乘法运算符实现。输入为两个32位整数，输出为两个32位整数，分别存放在LO和HI寄存器中。

考虑到乘法运算需要的时间比较长，为尽可能简化流水线设计，可适当降低时钟频率，以使得乘法运算可以在一个时钟周期内完成运算。

### 2.1.3 寄存器堆

寄存器堆负责实现通用寄存器的读写和在数据通路中的控制，在流水线译码阶段读取一个或两个通用寄存器的数据（组合逻辑），并在流水线写回阶段将结果写入通用寄存器（时序逻辑）。

寄存器堆采用FPGA的逻辑单元来实现数据的存储，在32位MIPS架构下需要实现32个32位通用寄存器。

### 2.1.4 CP0

### 2.1.5 异常中断处理

### 2.1.6 MMU

#### 2.1.6.1 虚拟地址映射

#### 2.1.6.2 TLB

## 2.2 Ucore

### 2.2.1 BIOS

BIOS即为启动ucore所用的Bootloader程序，通常是放在Flash中。而本实验中我们将在FPGA里建立一块ROM，将Bootloader放置在该ROM中，并且设置CPU的访问地址从该ROM开始。这样能避免由于Flash的读写不稳定而对BIOS造成的破坏，还能将ucore与其独立开来。

BIOS启动时，Flash中的操作系统加载到内存中，然后跳转到操作系统的初始化代码，从而开始操作系统的工作。

### 2.2.2 远程文件执行

实验要求修改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。

## 2.3 外设

### 2.3.1 串口

串口的功能需求为实现与PC机的通信，通过计算机键盘输入数据，向计算机输出数据。

串口模块的主要部分位于板子上的CPLD中，在FPGA端，对串口的控制通过data\_ready、tbre、tsre、rdn、wrn进行。

当data\_ready= '1' 时串口数据就绪，可以读出。

当tbre and tsre= '1' 是，表示可以向串口写入。

将rdn置 '0' 且数据线写高阻，可以从数据线得出串口数据。

将数据写入数据线且将wrn置 '0'，可以向串口发出数据。

串口模块1FD003F8地址表示数据（只有低8位有效），必须使用SW指令写入。

1FD003FC地址表示状态寄存器Status，Status&1=1时可写，Status&2=2时可读。

### 2.3.2 VGA

Device.VGA模块是显示控制模块，接受CPU写入的ASCII码数据，并维护一个字符矩阵，以字符中断的形式通过VGA接口输出到显示器。

CPU需要从1FC03000地址读出数据，如果结果为 '1' 表示可以写入，否则不能吸入；可以写入时，向1FC03000地址写入ASCII数据即可。

### 2.3.3 ps/2键盘

Device.Keyboard模块位键盘控制模块。只要从0F000000地址读出数据，结果为0表示没有新数据，否则读到的就是键入的ASCII码。

PS/2键盘读到的是扫描码，为了方便软件，可以在硬件层面加入编码转换，CPU读到的直接就是ASCII码。

### 2.3.4 网口

这是拓展要求。实验提供了DM9000A网口芯片与PC机进行网络通讯。该芯片带有通用处理器接口的以太网控制器，一个10/100M PHY和4K双字的SRAM，IO端口支持3.3V与5V容限值。

当收到数据包时，芯片通过中断信号方式通知CPU触发异常，由操作系统对数据包进行处理。若要完成此需求，则需对操作系统进行改写：在初始化时添加网口中断使能、添加网口芯片初始化代码、手动实现网络通信协议和网口驱动等。

## 2.4 Decaf编译器

### 2.4.1 汇编指令的生成

由于简化的CPU中并未实现add、sub指令，需要把decaf的MIPS后端里生成add、sub指令的部分改成addu、subu，区别仅在于溢出时后者不会产生异常。

另外，CPU中也未实现除法指令，不过由于所用测试程序中没有除法运算，因此不进行相关修改。如果需要除法，可以用其它指令手动实现除法函数，并把除法翻译成函数调用。

### 2.4.2 库函数调用及calling convention

标准MIPS32使用032 ABI，函数调用的前四个参数通过\$a0-\$a3四个寄存器传输；但decaf编译出的程序的参数全都在栈上传递。当然，无论什么calling convention，只要能自恰，程序本身就应该能正常运行，所以需要解决的问题只有用于程序与C实现的库函数及操作系统交互的部分。

在这里，我们在decaf和库函数之间增加一个适配器层，将decaf的调用约定翻译成032 ABI再调用库函数。

### 2.4.3 程序入口及退出

直接使用ucore里的linker script(user.ld)以及用户静态函数库libuser.a，在该环境下系统会设置好一些全局变量，然后跳转到main执行。我们将修改decaf编译器，将其输出的main重命名为decaf\_main，然后汇编实现一个新的main函数。由于decaf的main是void类型，我们默认其执行成功返回0，于是在decaf\_main返回后直接调用exit(0)。

## 2.5 指令集与数据通路

## 3 性能需求

## 4 运行环境需求

### 4.1 设备

### 4.2 控制

## 5 附录

### 5.1 指令系统

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] < Zero-extend(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 + Sign-extend(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] >= 0) PC $\leftarrow$ PC + Sign-extend(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 + Sign-extend(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] <= 0) PC $\leftarrow$ PC + Sign-extend(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 + Sign-extend(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] != R[t]) PC $\leftarrow$ PC + Sign-extend(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 + Sign-extend(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 + Sign-extend(immediate), RA $\leftarrow$ RPC															
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	$R[t] \leftarrow \text{MEM}[R[s] + \text{Sign-extend}(\text{immediate})]$															
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	$\text{MEM}[R[s] + \text{Sign-extend}(\text{immediate})] \leftarrow R[t]$															
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	$R[t] \leftarrow R[s] \& \text{Zero-extend}(\text{immediate})$															
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	$R[d] \leftarrow \sim(R[s] \mid R[t])$															
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	$R[d] \leftarrow R[s] \mid R[t]$															
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	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
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	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	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	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															