

## 3.2 指令功能分类

处理器需要实现的指令包括除 4 条非对齐指令外的所有 MIPS I 指令以及 MIPS32 中的 ERET 指令，有 14 条算术运算指令、8 条逻辑运算指令，6 条移位指令、8 条分支跳转指令、4 条数据移动指令、2 条自陷指令、12 条访存指令、3 条特权指令，共计 57 条。下面分类给出各部分指令的简要功能介绍。

表 3-1 算术运算指令

指令名称格式	指令功能简述
ADD rd, rs, rt	加（可产生溢出例外）
ADDI rt, rs, immediate	加立即数（可产生溢出例外）
ADDU rd, rs, rt	加（不产生溢出例外）
ADDIU rt, rs, immediate	加立即数（不产生溢出例外）
SUB rd, rs, rt	减（可产生溢出例外）
SUBU rd, rs, rt	减（不产生溢出例外）
SLT rd, rs, rt	有符号小于置 1
SLTI rt, rs, immediate	有符号小于立即数设置 1
SLTU rd, rs, rt	无符号小于设置 1
SLTIU rt, rs, immediate	无符号小于立即数 <sup>[1]</sup> 设置 1
DIV rs, rt	有符号字除
DIVU rs, rt	无符号字除
MULT rs, rt	有符号字乘
MULTU rs, rt	无符号字乘

表 3-2 逻辑运算指令

指令名称格式	指令功能简述
AND rd, rs, rt	位与
ANDI rt, rs, immediate	立即数位与
LUI rt, immediate	寄存器高半部分置立即数
NOR rd, rs, rt	位或非
OR rd, rs, rt	位或
ORI rt, rs, immediate	立即数位或

<sup>[1]</sup> 请注意虽然是无符号比较，但是立即数仍是进行有符号扩展。

指令名称格式	指令功能简述
XOR rd, rs, rt	位异或
XORI rt, rs, immediate	立即数位异或

表 3-3 移位指令

指令名称格式	指令功能简述
SLL rd, rt, sa	立即数逻辑左移
SLLV rd, rs, rt	变量逻辑左移
SRA rd, rt, sa	立即数算术右移
SRAV rd, rs, rt	变量算术右移
SRL rd, rt, sa	立即数逻辑右移
SRLV rd, rs, rt	变量逻辑右移

表 3-4 分支跳转指令

指令名称格式	指令功能简述
BEQ rs, rt, offset	相等转移
BNE rs, rt, offset	不等转移
BGEZ rs, offset	大于等于 0 转移
BGTZ rs, offset	大于 0 转移
BLEZ rs, offset	小于等于 0 转移
BLTZ rs, offset	小于 0 转移
BLTZAL rs, offset	小于 0 调用子程序并保存返回地址
BGEZAL rs, offset	大于等于 0 调用子程序并保存返回地址
J target	无条件直接跳转
JAL target	无条件直接跳转至子程序并保存返回地址
JR rs	无条件寄存器跳转
JALR rd, rs	无条件寄存器跳转至子程序并保存返回地址下

表 3-5 数据移动指令

指令名称格式	指令功能简述
MFHI rd	HI 寄存器至通用寄存器
MFLO rd	L0 寄存器至通用寄存器
MTHI rs	通用寄存器至 HI 寄存器
MTLO rs	通用寄存器至 L0 寄存器

表 3-6 自陷指令

指令名称格式	指令功能简述
BREAK	断点
SYSCALL	系统调用

表 3-7 访存指令

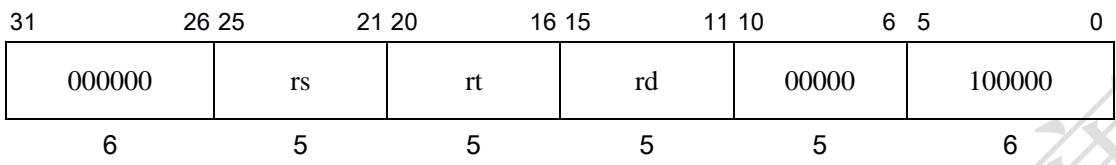
指令名称格式	指令功能简述
LB rt, offset(base)	取字节有符号扩展
LBU rt, offset(base)	取字节无符号扩展
LH rt, offset(base)	取半字有符号扩展
LHU rt, offset(base)	取半字无符号扩展
LW rt, offset(base)	取字
SB rt, offset(base)	存字节
SH rt, offset(base)	存半字
SW rt, offset(base)	存字

表 3-8 自陷指令

指令名称格式	指令功能简述
ERET	例外处理返回
MFC0	读 CP0 寄存器值至通用寄存器
MTC0	通用寄存器值写入 CP0 寄存器

### 3.3 算术运算指令

#### 3.3.1 ADD



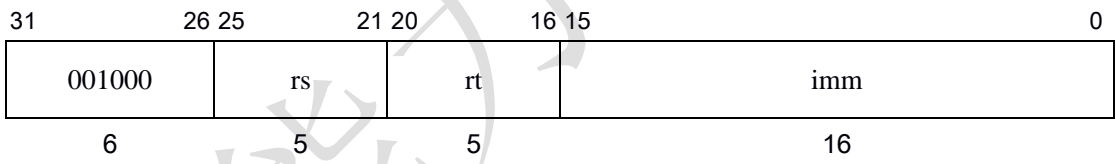
汇编格式: ADD rd, rs, rt

功能描述: 将寄存器 rs 的值与寄存器 rt 的值相加, 结果写入寄存器 rd 中。如果产生溢出, 则触发整型溢出例外 (IntegerOverflow)。

操作定义:  $tmp \leftarrow (GPR[rs]_{31:0} + GPR[rt]_{31:0})$   
if  $tmp_{32} \neq tmp_{31}$  then  
    SignalException(IntegerOverflow)  
else  
     $GPR[rd] \leftarrow tmp_{31:0}$   
endif

例 外: 如果有溢出, 则触发整型溢出例外。

#### 3.3.2 ADDI



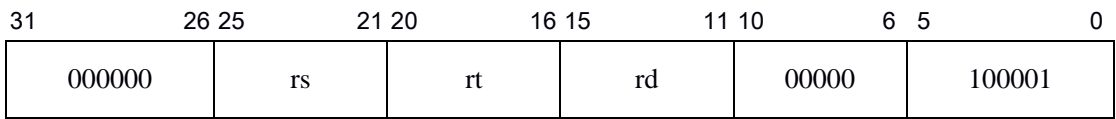
汇编格式: ADDI rt, rs, imm

功能描述: 将寄存器 rs 的值与有符号扩展至 32 位的立即数 imm 相加, 结果写入 rt 寄存器中。如果产生溢出, 则触发整型溢出例外 (IntegerOverflow)。

操作定义:  $tmp \leftarrow (GPR[rs]_{31:0} + sign\_extend(imm))$   
if  $tmp_{32} \neq tmp_{31}$  then  
    SignalException(IntegerOverflow)  
else  
     $GPR[rt] \leftarrow tmp_{31:0}$   
endif

例 外: 如果有溢出, 则触发整型溢出例外。

#### 3.3.3 ADDU



6 5 5 5 5 6

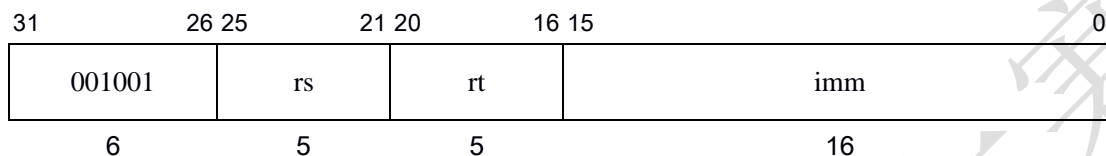
汇编格式: ADDU rd, rs, rt

功能描述: 将寄存器 rs 的值与寄存器 rt 的值相加, 结果写入 rd 寄存器中。

操作定义:  $GPR[rd] \leftarrow GPR[rs] + GPR[rt]$

例 外: 无

### 3.3.4 ADDIU



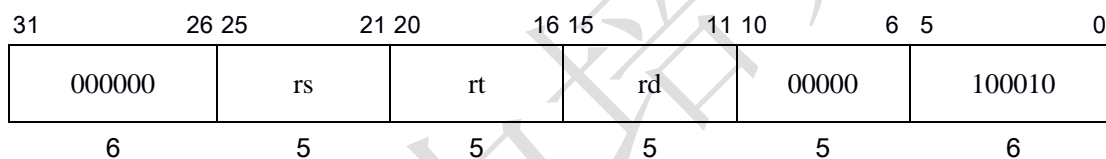
汇编格式: ADDIU rt, rs, imm

功能描述: 将寄存器 rs 的值与有符号扩展至 32 位的立即数 imm 相加, 结果写入 rt 寄存器中。

操作定义:  $GPR[rt] \leftarrow GPR[rs] + \text{sign\_extend}(imm)$

例 外: 无

### 3.3.5 SUB



汇编格式: SUB rd, rs, rt

功能描述: 将寄存器 rs 的值与寄存器 rt 的值相减, 结果写入 rd 寄存器中。如果产生溢出, 则触发整型溢出例外 (IntegerOverflow)。

操作定义:  $tmp \leftarrow (GPR[rs]_{31:0} - GPR[rt]_{31:0})$

if  $tmp_{32} \neq tmp_{31}$  then

SignalException(IntegerOverflow)

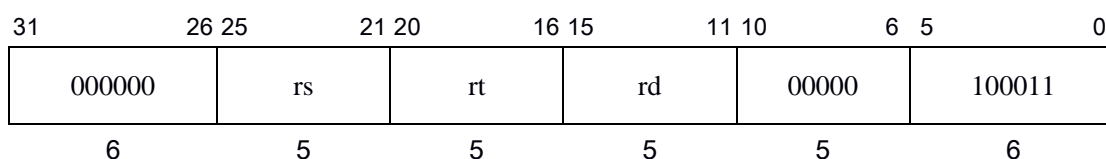
else

$GPR[rd] \leftarrow tmp_{31:0}$

endif

例 外: 如果有溢出, 则触发整型溢出例外。

### 3.3.6 SUBU



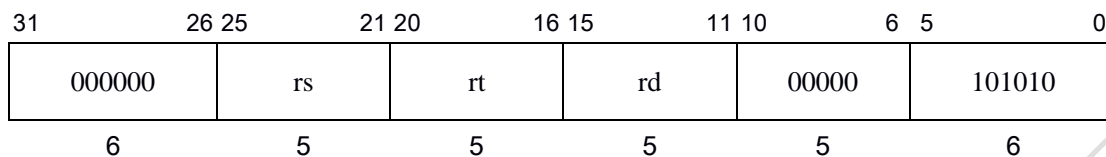
汇编格式: SUBU rd, rs, rt

功能描述: 将寄存器 rs 的值与寄存器 rt 的值相减, 结果写入 rd 寄存器中。

操作定义:  $GPR[rd] \leftarrow GPR[rs] - GPR[rt]$

例 外: 无

### 3.3.7 SLT



汇编格式: SLT rd, rt, rs

功能描述: 将寄存器 rs 的值与寄存器 rt 中的值进行有符号数比较, 如果寄存器 rs 中的值小, 则寄存器 rd 置 1; 否则寄存器 rd 置 0。

操作定义: if  $GPR[rs] < GPR[rt]$  then

$GPR[rd] \leftarrow 1$

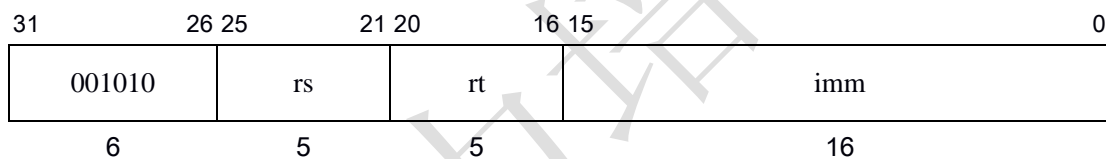
else

$GPR[rd] \leftarrow 0$

endif

例 外: 无

### 3.3.8 SLTI



汇编格式: SLTI rt, rs, imm

功能描述: 将寄存器 rs 的值与有符号扩展至 32 位的立即数 imm 进行有符号数比较, 如果寄存器 rs 中的值小, 则寄存器 rt 置 1; 否则寄存器 rt 置 0。

操作定义: if  $GPR[rs] < \text{Sign\_extend}(imm)$  then

$GPR[rt] \leftarrow 1$

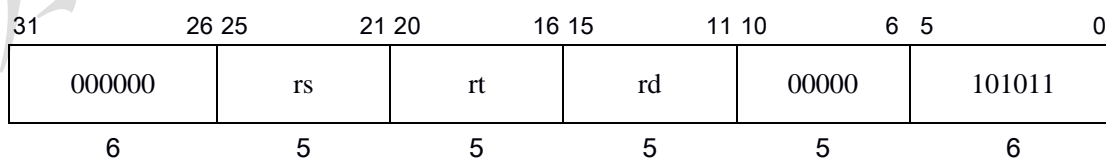
else

$GPR[rt] \leftarrow 0$

endif

例 外: 无

### 3.3.9 SLTU



汇编格式: SLTU rd, rs, rt

功能描述: 将寄存器 rs 的值与寄存器 rt 中的值进行无符号数比较, 如果寄存器 rs 中的值小, 则寄存器 rd 置 1; 否则寄存器 rd 置 0。

操作定义: if  $(0 \| GPR[rs]_{31..0}) < (0 \| GPR[rt]_{31..0})$  then

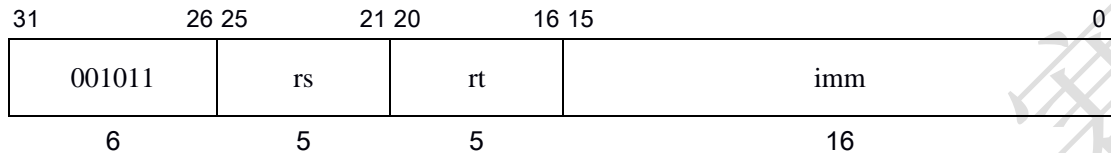
```

        GPR[rd] ← 1
    else
        GPR[rd] ← 0
    endif

```

例 外：无

### 3.3.10 SLTIU



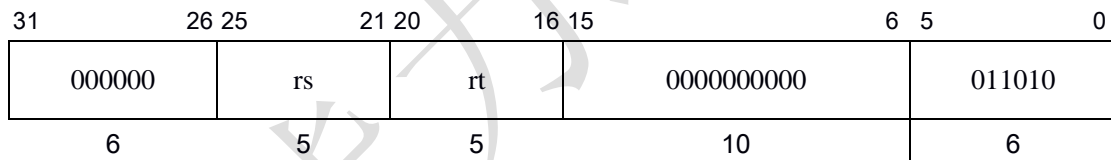
汇编格式：SLTIU rt, rs, imm

功能描述：将寄存器 rs 的值与有符号扩展至 32 位的立即数 imm 进行无符号数比较，如果寄存器 rs 中的值小，则寄存器 rt 置 1；否则寄存器 rt 置 0。

操作定义：if  $(0 \| \text{GPR}[\text{rs}]_{31..0}) < \text{Sign\_extend}(\text{imm})$  then  
              $\text{GPR}[\text{rt}] \leftarrow 1$   
 else  
              $\text{GPR}[\text{rt}] \leftarrow 0$   
 endif

例 外：无

### 3.3.11 DIV



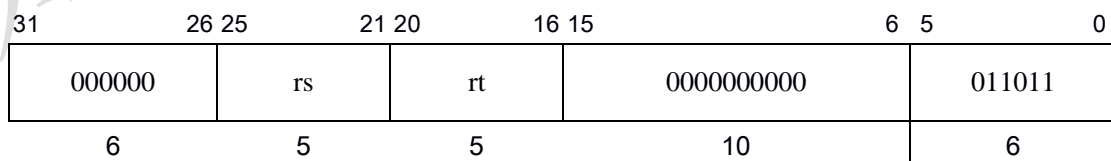
汇编格式：DIV rs, rt

功能描述：有符号除法，寄存器 rs 的值除以寄存器 rt 的值，商写入 LO 寄存器中，余数写入 HI 寄存器中。

操作定义： $q \leftarrow \text{GPR}[\text{rs}]_{31..0} \text{ div } \text{GPR}[\text{rt}]_{31..0}$   
 $\text{LO} \leftarrow q$   
 $r \leftarrow \text{GPR}[\text{rs}]_{31..0} \text{ mod } \text{GPR}[\text{rt}]_{31..0}$   
 $\text{HI} \leftarrow r$

例 外：无

### 3.3.12 DIVU



汇编格式：DIVU rs, rt

功能描述：无符号除法，寄存器 rs 的值除以寄存器 rt 的值，商写入 LO 寄存器中，余数写入 HI 寄存器中。

操作定义： $q \leftarrow (0 \| \text{GPR}[\text{rs}]_{31..0}) \text{ div } (0 \| \text{GPR}[\text{rt}]_{31..0})$   
 $\text{LO} \leftarrow q$

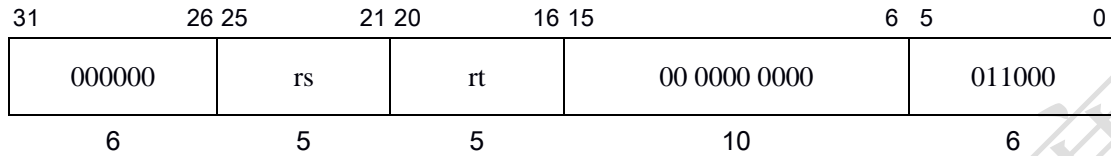
---


$$r \leftarrow (0 \parallel \text{GPR}[\text{rs}]_{31..0}) \bmod (0 \parallel \text{GPR}[\text{rt}]_{31..0})$$

$$\text{HI} \leftarrow r$$

例 外：无

### 3.3.13 MULT



汇编格式：MULT rs, rt

功能描述：有符号乘法，寄存器 rs 的值乘以寄存器 rt 的值，乘积的低半部分和高半部分分别写入 LO 寄存器 and HI 寄存器。

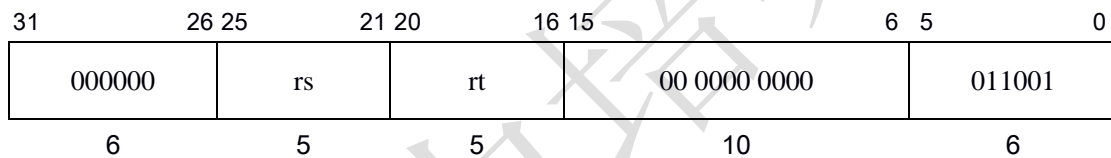
操作定义： $\text{prod} \leftarrow \text{GPR}[\text{rs}]_{31..0} \times \text{GPR}[\text{rt}]_{31..0}$

$$\text{LO} \leftarrow \text{prod}_{31..0}$$

$$\text{HI} \leftarrow \text{prod}_{63..32}$$

例 外：无

### 3.3.14 MULTU



汇编格式：MULTU rs, rt

功能描述：无符号乘法，寄存器 rs 的值乘以寄存器 rt 的值，乘积的低半部分和高半部分分别写入 LO 寄存器 and HI 寄存器。

操作定义： $\text{prod} \leftarrow (0 \parallel \text{GPR}[\text{rs}]_{31..0}) \times (0 \parallel \text{GPR}[\text{rt}]_{31..0})$

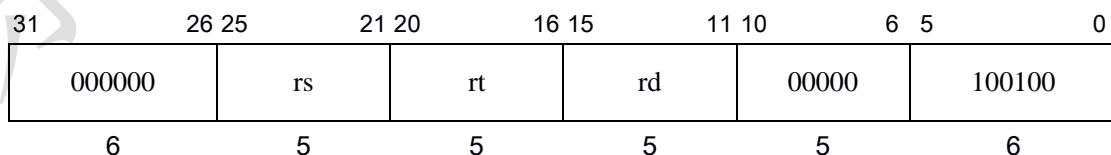
$$\text{LO} \leftarrow \text{prod}_{31..0}$$

$$\text{HI} \leftarrow \text{prod}_{63..32}$$

例 外：无

## 3.4 逻辑运算指令

### 3.4.1 AND



汇编格式：AND rd, rs, rt

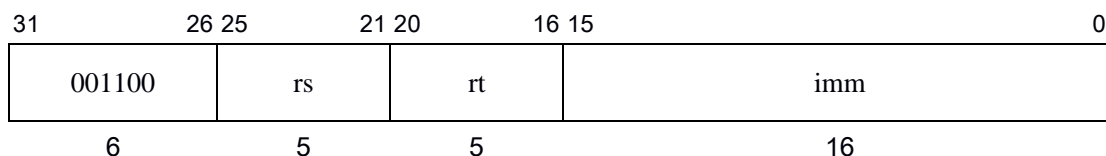
功能描述：寄存器 rs 中的值与寄存器 rt 中的值按位逻辑与，结果写入寄存器 rd 中。

操作定义： $\text{GPR}[\text{rd}] \leftarrow \text{GPR}[\text{rs}] \& \text{GPR}[\text{rt}]$

例 外：无



### 3.4.2 ANDI



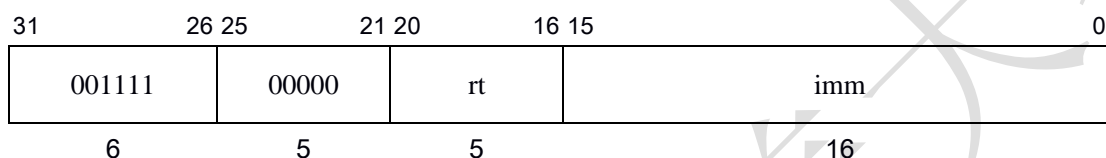
汇编格式: ANDI rt, rs, imm

功能描述: 寄存器 rs 中的值与 0 扩展至 32 位的立即数 imm 按位逻辑与, 结果写入寄存器 rt 中。

操作定义:  $GPR[rt] \leftarrow GPR[rs] \text{ and Zero\_extend}(imm)$

例 外: 无

### 3.4.3 LUI



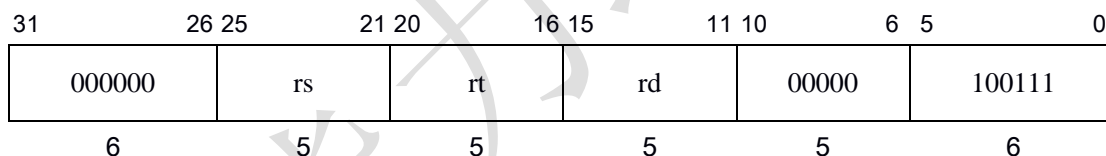
汇编格式: LUI rt, imm

功能描述: 将 16 位立即数 imm 写入寄存器 rt 的高 16 位, 寄存器 rt 的低 16 位置 0。

操作定义:  $GPR[rt] \leftarrow (imm \parallel 0^{16})$

例 外: 无

### 3.4.4 NOR



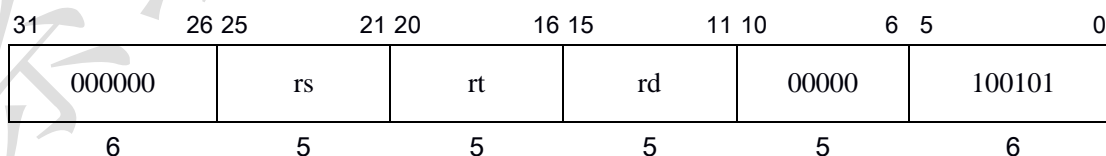
汇编格式: NOR rd, rs, rt

功能描述: 寄存器 rs 中的值与寄存器 rt 中的值按位逻辑或非, 结果写入寄存器 rd 中。

操作定义:  $GPR[rd] \leftarrow GPR[rs] \text{ nor } GPR[rt]$

例 外: 无

### 3.4.5 OR



汇编格式: OR rd, rs, rt

功能描述: 寄存器 rs 中的值与寄存器 rt 中的值按位逻辑或, 结果写入寄存器 rd 中。

操作定义:  $GPR[rd] \leftarrow GPR[rs] \text{ or } GPR[rt]$

例 外: 无

### 3.4.6 ORI

31	26 25	21 20	16 15	0
001101	rs	rt	imm	
6	5	5	16	

汇编格式: ORI rt, rs, imm

功能描述: 寄存器 rs 中的值与 0 扩展至 32 位的立即数 imm 按位逻辑或, 结果写入寄存器 rt 中。

操作定义:  $GPR[rt] \leftarrow GPR[rs] \text{ or } Zero\_extend(imm)$

例 外: 无

### 3.4.7 XOR

31	26 25	21 20	16 15	11 10	6 5	0
000000	rs	rt	rd	00000	100110	
6	5	5	5	5	6	

汇编格式: XOR rd, rs, rt

功能描述: 寄存器 rs 中的值与寄存器 rt 中的值按位逻辑异或, 结果写入寄存器 rd 中。

操作定义:  $GPR[rd] \leftarrow GPR[rs] \text{ xor } GPR[rt]$

例 外: 无

### 3.4.8 XORI

31	26 25	21 20	16 15	0
001110	rs	rt	imm	
6	5	5	16	

汇编格式: XORI rt, rs, imm

功能描述: 寄存器 rs 中的值与 0 扩展至 32 位的立即数 imm 按位逻辑异或, 结果写入寄存器 rt 中。

操作定义:  $GPR[rt] \leftarrow GPR[rs] \text{ xor } Zero\_extend(imm)$

例 外: 无

## 3.5 移位指令

### 3.5.1 SLLV

31	26 25	21 20	16 15	11 10	6 5	0
000000	rs	rt	rd	00000	000100	
6	5	5	5	5	6	

汇编格式: SLLV rd, rt, rs

功能描述: 由寄存器 rs 中的值指定移位置, 对寄存器 rt 的值进行逻辑左移, 结果写入寄存器 rd 中。

操作定义:  $s \leftarrow GPR[rs]_{4..0}$   
 $GPR[rd] \leftarrow GPR[rt]_{(31-s)..0} || 0^s$

例 外：无

### 3.5.2 SLL

31	26 25	21 20	16 15	11 10	6 5	0
000000	00000	rt	rd	sa	000000	
6	5	5	5	5	6	

汇编格式：SLL rd, rt, sa

功能描述：由立即数 sa 指定移位量，对寄存器 rt 的值进行逻辑左移，结果写入寄存器 rd 中。

操作定义：s ← sa

$GPR[rd] \leftarrow GPR[rt]_{(31-s)..0} \parallel 0^s$

例 外：无

### 3.5.3 SRAV

31	26 25	21 20	16 15	11 10	6 5	0
000000	rs	rt	rd	00000	000111	
6	5	5	5	5	6	

汇编格式：SRAV rd, rt, rs

功能描述：由寄存器 rs 中的值指定移位量，对寄存器 rt 的值进行算术右移，结果写入寄存器 rd 中。

操作定义：s ← GPR[rs]<sub>4..0</sub>

$GPR[rd] \leftarrow (GPR[rt]_{31})^s \parallel GPR[rt]_{31..s}$

例 外：无

### 3.5.4 SRA

31	26 25	21 20	16 15	11 10	6 5	0
000000	00000	rt	rd	sa	000011	
6	5	5	5	5	6	

汇编格式：SRA rd, rt, sa

功能描述：由立即数 sa 指定移位量，对寄存器 rt 的值进行算术右移，结果写入寄存器 rd 中。

操作定义：s ← sa

$GPR[rd] \leftarrow (GPR[rt]_{31})^s \parallel GPR[rt]_{31..s}$

例 外：无

### 3.5.5 SRLV

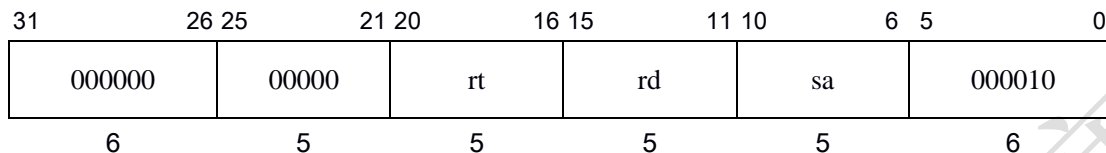
31	26 25	21 20	16 15	11 10	6 5	0
000000	rs	rt	rd	00000	000110	
6	5	5	5	5	6	

汇编格式：SRLV rd, rt, rs

功能描述：由寄存器 rs 中的值指定移位量，对寄存器 rt 的值进行逻辑右移，结果写入寄存器 rd 中。

操作定义:  $s \leftarrow \text{GPR}[\text{rs}]_{4..0}$   
 $\text{GPR}[\text{rd}] \leftarrow 0^s \parallel \text{GPR}[\text{rt}]_{31..s}$   
 例 外: 无

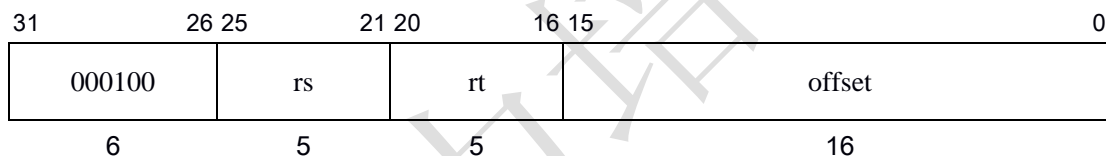
### 3.5.6 SRL



汇编格式: SRL rd, rt, sa  
 功能描述: 由立即数 sa 指定移位量, 对寄存器 rt 的值进行逻辑右移, 结果写入寄存器 rd 中。  
 操作定义:  $s \leftarrow \text{sa}$   
 $\text{GPR}[\text{rd}] \leftarrow 0^s \parallel \text{GPR}[\text{rt}]_{31..s}$   
 例 外: 无

## 3.6 分支跳转指令

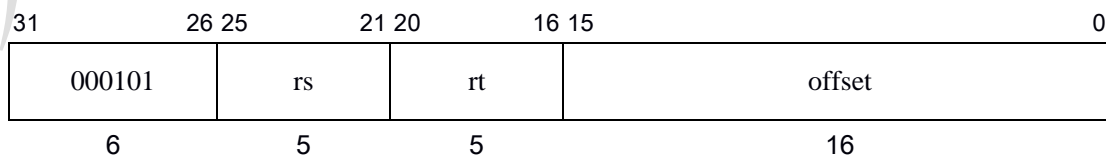
### 3.6.1 BEQ



汇编格式: BEQ rs, rt, offset  
 功能描述: 如果寄存器 rs 的值等于寄存器 rt 的值则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。  
 操作定义: I:  $\text{condition} \leftarrow \text{GPR}[\text{rs}] = \text{GPR}[\text{rt}]$   
 $\text{target\_offset} \leftarrow \text{Sign\_extend}(\text{offset} \ll 2)$   
 I+1: if condition then  
 $\text{PC} \leftarrow \text{PC} + \text{target\_offset}$   
 endif

例 外: 无

### 3.6.2 BNE

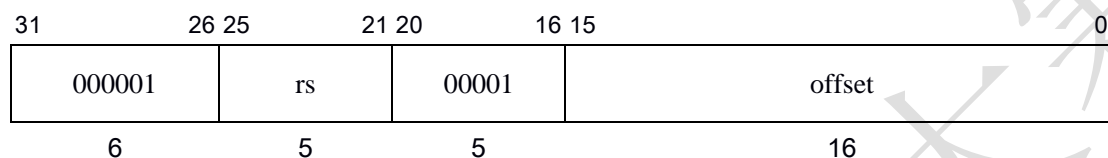


汇编格式: BNE rs, offset  
 功能描述: 如果寄存器 rs 的值不等于寄存器 rt 的值则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。

操作定义: I:      $\text{condition} \leftarrow \text{GPR}[\text{rs}] \neq \text{GPR}[\text{rt}]$   
                    $\text{target\_offset} \leftarrow \text{Sign\_extend}(\text{offset}||0^2)$   
           I+1:   if condition then  
                    $\text{PC} \leftarrow \text{PC} + \text{target\_offset}$   
                   endif

例 外: 无

### 3.6.3 BGEZ



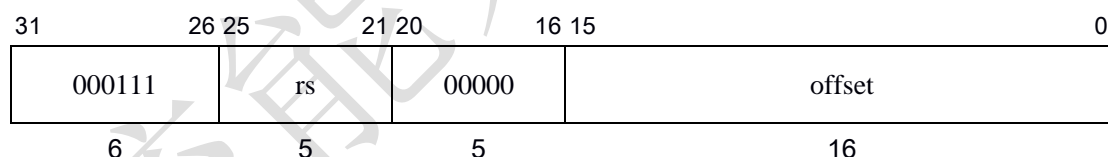
汇编格式: BGEZ rs, offset

功能描述: 如果寄存器 rs 的值大于等于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。

操作定义: I:      $\text{condition} \leftarrow \text{GPR}[\text{rs}] \geq 0$   
                    $\text{target\_offset} \leftarrow \text{Sign\_extend}(\text{offset}||0^2)$   
           I+1:   if condition then  
                    $\text{PC} \leftarrow \text{PC} + \text{target\_offset}$   
                   endif

例 外: 无

### 3.6.4 BGTZ



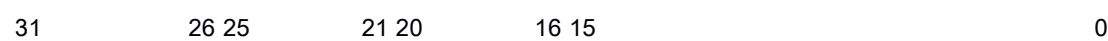
汇编格式: BGTZ rs, offset

功能描述: 如果寄存器 rs 的值大于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。

操作定义: I:      $\text{condition} \leftarrow \text{GPR}[\text{rs}] > 0$   
                    $\text{target\_offset} \leftarrow \text{Sign\_extend}(\text{offset}||0^2)$   
           I+1:   if condition then  
                    $\text{PC} \leftarrow \text{PC} + \text{target\_offset}$   
                   endif

例 外: 无

### 3.6.5 BLEZ



000110	rs	00000	offset
6	5	5	16

汇编格式: BLEZ rs, offset

功能描述: 如果寄存器 rs 的值小于等于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。

操作定义: I:      $condition \leftarrow GPR[rs] \leq 0$   
                    $target\_offset \leftarrow Sign\_extend(offset \ll 2)$   
   I+1:   if condition then  
              $PC \leftarrow PC + target\_offset$   
           endif

例 外: 无

### 3.6.6 BLTZ

31	26 25	21 20	16 15	0
000001	rs	00000	offset	
6	5	5	16	

汇编格式: BLTZ rs, offset

功能描述: 如果寄存器 rs 的值小于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。

操作定义: I:      $condition \leftarrow GPR[rs] < 0$   
                    $target\_offset \leftarrow Sign\_extend(offset \ll 2)$   
   I+1:   if condition then  
              $PC \leftarrow PC + target\_offset$   
           endif

例 外: 无

### 3.6.7 BGEZAL

31	26 25	21 20	16 15	0
000001	rs	10001	offset	
6	5	5	16	

汇编格式: BGEZAL rs, offset

功能描述: 如果寄存器 rs 的值大于等于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。无论转移与否, 将该分支对应延迟槽指令之后的指令的 PC 值保存至第 31 号通用寄存器中。

操作定义: I:      $condition \leftarrow GPR[rs] \geq 0$   
                    $target\_offset \leftarrow Sign\_extend(offset \ll 2)$

---

$\text{GPR}[31] \leftarrow \text{PC} + 8$

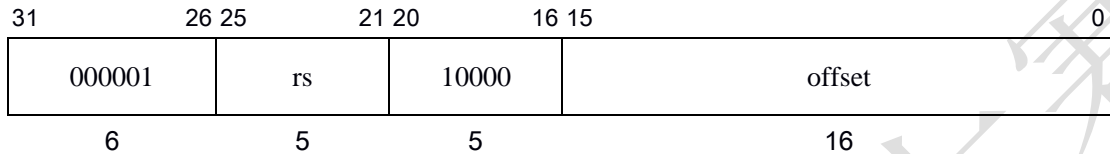
I+1: if condition then

$\text{PC} \leftarrow \text{PC} + \text{target\_offset}$

endif

例 外: 无

### 3.6.8 BLTZAL



汇编格式: BLTZAL rs, offset

功能描述: 如果寄存器 rs 的值小于 0 则转移, 否则顺序执行。转移目标由立即数 offset 左移 2 位并进行有符号扩展的值加上该分支指令对应的延迟槽指令的 PC 计算得到。无论转移与否, 将该分支对应延迟槽指令之后的指令的 PC 值保存至第 31 号通用寄存器中。

操作定义: I:  $\text{condition} \leftarrow \text{GPR}[\text{rs}] < 0$

$\text{target\_offset} \leftarrow \text{Sign\_extend}(\text{offset} \ll 0^2)$

$\text{GPR}[31] \leftarrow \text{PC} + 8$

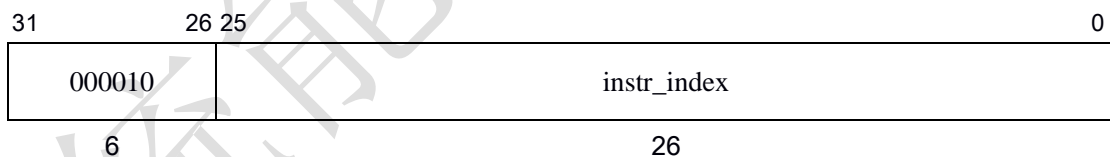
I+1: if condition then

$\text{PC} \leftarrow \text{PC} + \text{target\_offset}$

endif

例 外: 无

### 3.6.9 J



汇编格式: J target

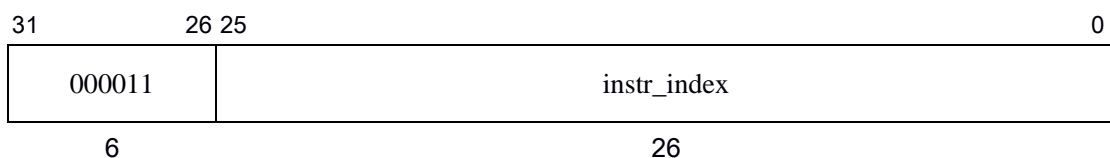
功能描述: 无条件跳转。跳转目标由该分支指令对应的延迟槽指令的 PC 的最高 4 位与立即数 instr\_index 左移 2 位后的值拼接得到。

操作定义: I:

I+1:  $\text{PC} \leftarrow \text{PC}_{31..28} \parallel \text{instr\_index} \parallel 0^2$

例 外: 无

### 3.6.10 JAL



汇编格式: JAL target

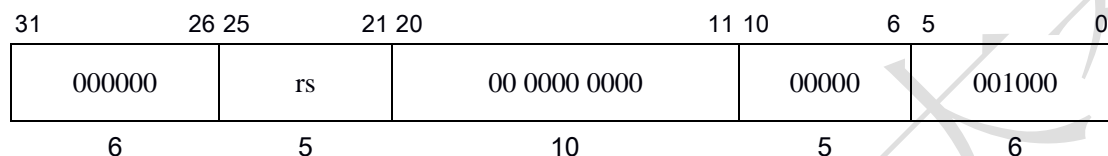
功能描述: 无条件跳转。跳转目标由该分支指令对应的延迟槽指令的 PC 的最高 4 位与立即数 instr\_index 左移 2 位后的值拼接得到。同时将该分支对应延迟槽指令之后的指令的 PC 值保存至第 31 号通用寄存器中。

操作定义: I:  $GPR[31] \leftarrow PC + 8$

I+1:  $PC \leftarrow PC_{31..28} \parallel instr\_index \parallel 0^2$

例 外: 无

### 3.6.11 JR



汇编格式: JR rs

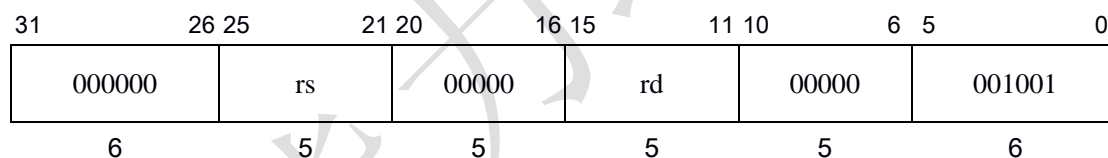
功能描述: 无条件跳转。跳转目标为寄存器 rs 中的值。

操作定义: I:  $temp \leftarrow GPR[rs]$

I+1:  $PC \leftarrow temp$

例 外: 无

### 3.6.12 JALR



汇编格式: JALR rd, rs

JALR rs (rd=31 implied)

功能描述: 无条件跳转。跳转目标为寄存器 rs 中的值。同时将该分支对应延迟槽指令之后的指令的 PC 值保存至寄存器 rd 中。

操作定义: I:  $temp \leftarrow GPR[rs]$

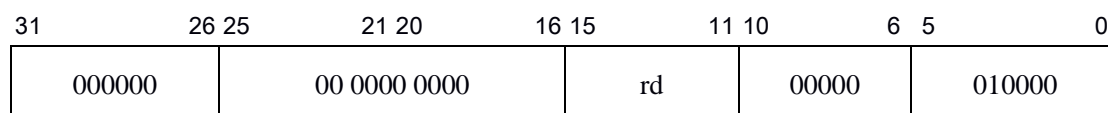
$GPR[rd] \leftarrow PC + 8$

I+1:  $PC \leftarrow temp$

例 外: 无

## 3.7 数据移动指令

### 3.7.1 MFHI





6 10 5 5 6

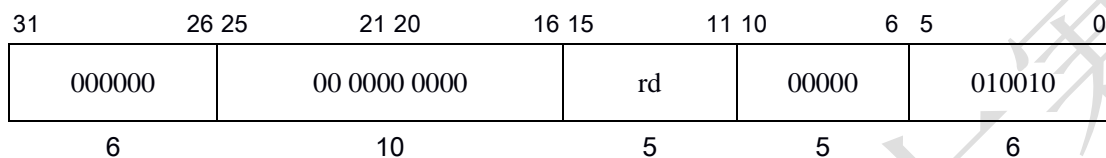
汇编格式: MFHI rd

功能描述: 将 HI 寄存器的值写入到寄存器 rd 中。

操作定义:  $GPR[rd] \leftarrow HI$

例 外: 无

### 3.7.2 MFLO



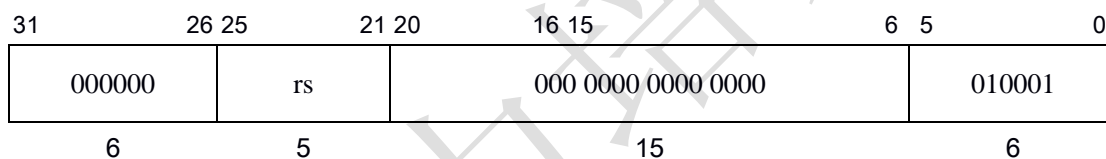
汇编格式: MFLO rd

功能描述: 将 LO 寄存器的值写入到寄存器 rd 中。

操作定义:  $GPR[rd] \leftarrow LO$

例 外: 无

### 3.7.3 MTHI



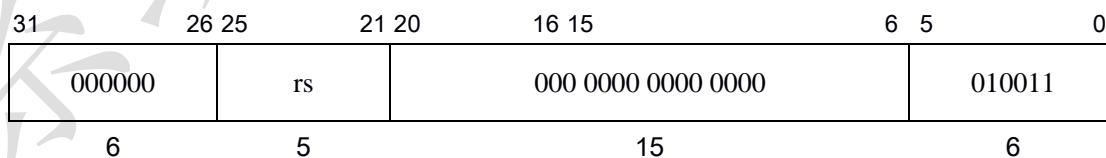
汇编格式: MTHI rs

功能描述: 将寄存器 rs 的值写入到 HI 寄存器中。

操作定义:  $HI \leftarrow GPR[rs]$

例 外: 无

### 3.7.4 MTLO



汇编格式: MTLO rs

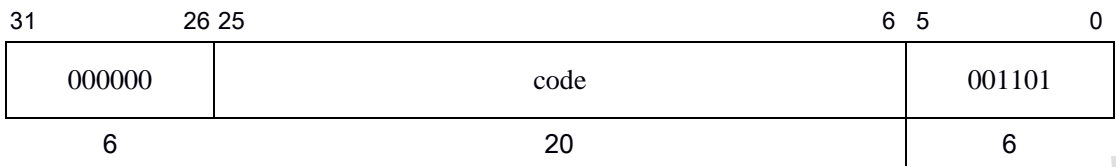
功能描述: 将寄存器 rs 的值写入到 LO 寄存器中。

操作定义:  $LO \leftarrow GPR[rs]$

例 外: 无

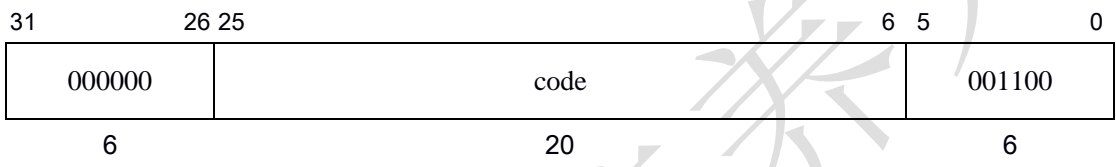
### 3.8 自陷指令

#### 3.8.1 BREAK



汇编格式: BREAK  
功能描述: 触发断点例外。  
操作定义: `SignalException(Breakpoint)`  
例 外: 断点例外

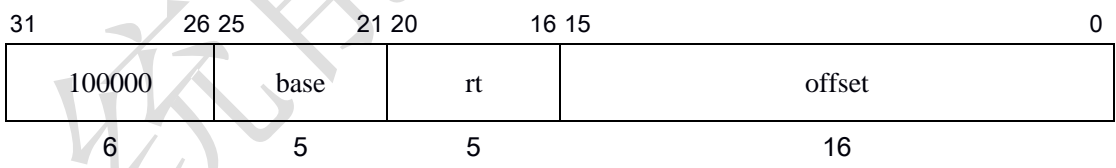
#### 3.8.2 SYSCALL



汇编格式: SYSCALL  
功能描述: 触发系统调用例外。  
操作定义: `SignalException(SystemCall)`  
例 外: 系统调用例外

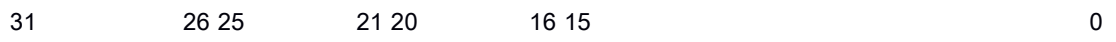
### 3.9 访存指令

#### 3.9.1 LB



汇编格式: `LB rt, offset(base)`  
功能描述: 将 `base` 寄存器的值加上符号扩展后的立即数 `offset` 得到访存的虚地址，据此虚地址从存储器中读取 1 个字节的值并进行符号扩展，写入到 `rt` 寄存器中。  
操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
 $(pAddr, CCA) \leftarrow AddressTranslation(vAddr, DATA, LOAD)$   
 $membyte \leftarrow LoadMemory(CCA, BYTE, pAddr, vAddr, DATA)$   
 $GPR[rt] \leftarrow sign\_extend(membyte_{7..0})$   
例 外: 无

#### 3.9.2 LBU



100100	base	rt	offset
6	5	5	16

汇编格式: LBU rt, offset(base)

功能描述: 将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址, 据此虚地址从存储器中读取 1 个字节的值并进行 0 扩展, 写入到 rt 寄存器中。

操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
 $(pAddr, CCA) \leftarrow AddressTranslation(vAddr, DATA, LOAD)$   
 $membyte \leftarrow LoadMemory(CCA, BYTE, pAddr, vAddr, DATA)$   
 $GPR[rt] \leftarrow zero\_extend(membyte_{7..0})$

例 外: 无

### 3.9.3 LH

31	26 25	21 20	16 15	0
100001	base	rt	offset	
6	5	5	16	

汇编格式: LH rt, offset(base)

功能描述: 将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址, 如果地址不是 2 的整数倍则触发地址错例外, 否则据此虚地址从存储器中读取连续 2 个字节的值并进行符号扩展, 写入到 rt 寄存器中。

操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
if  $vAddr_0 \neq 0$  then  
    SignalException(AddressError)  
endif  
 $(pAddr, CCA) \leftarrow AddressTranslation(vAddr, DATA, LOAD)$   
 $memhalf \leftarrow LoadMemory(CCA, HALFWORD, pAddr, vAddr, DATA)$   
 $GPR[rt] \leftarrow sign\_extend(memhalf_{15..0})$

例 外: 地址最低 1 位不为 0, 触发地址错例外

### 3.9.4 LHU

31	26 25	21 20	16 15	0
100101	base	rt	offset	
6	5	5	16	

汇编格式: LHU rt, offset(base)

功能描述: 将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址, 如果地址不是 2 的整数倍则触发地址错例外, 否则据此虚地址从存储器中读取连续 2 个字节的值并进行 0 扩展, 写入到 rt 寄存器中。

操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
if  $vAddr_0 \neq 0$  then  
    SignalException(AddressError)  
endif

---

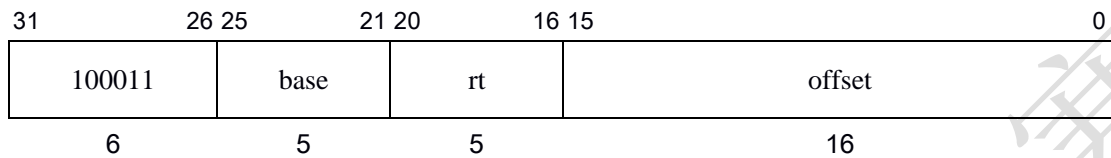
```

(pAddr, CCA) ← AddressTranslation(vAddr, DATA, LOAD)
memhalf ← LoadMemory(CCA, HALFWORD, pAddr, vAddr, DATA)
GPR[rt] ← zero_extend(memhalf15..0)

```

例 外：地址最低 1 位不为 0，触发地址错例外

### 3.9.5 LW



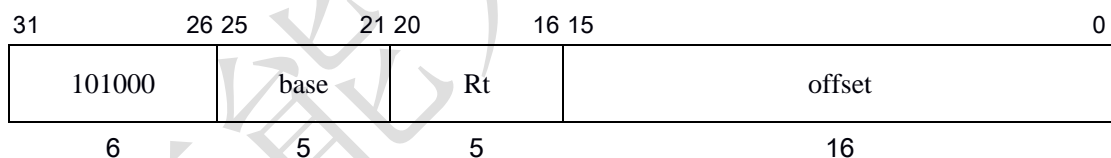
汇编格式：LW rt, offset(base)

**功能描述：**将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址，如果地址不是 4 的整数倍则触发地址错例外，否则据此虚地址从存储器中读取连续 4 个字节的值并进行符号扩展，写入到 rt 寄存器中。

**操作定义：** $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
 if  $vAddr_{1..0} \neq 0$  then  
     SignalException(AddressError)  
 endif  
 $(pAddr, CCA) \leftarrow AddressTranslation(vAddr, DATA, LOAD)$   
 $memword \leftarrow LoadMemory(CCA, WORD, pAddr, vAddr, DATA)$   
 $GPR[rt] \leftarrow sign\_extend(memword_{31..0})$

例 外：地址最低 2 位不为 0，触发地址错例外

### 3.9.6 SB



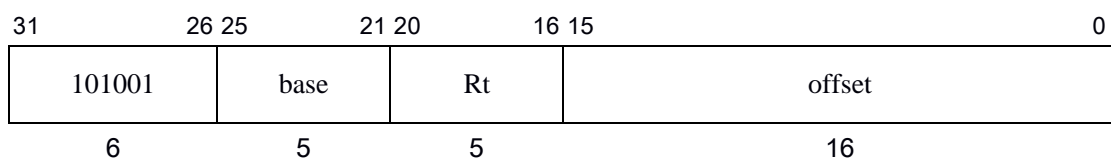
汇编格式：SB rt, offset(base)

**功能描述：**将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址，据此虚地址将 rt 寄存器的最低字节存入存储器中。

**操作定义：** $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
 $(pAddr, CCA) \leftarrow AddressTranslation(vAddr, DATA, STORE)$   
 $databyte \leftarrow GPR[rt]_{7..0}$   
 $StoreMemory(CCA, BYTE, databyte, pAddr, vAddr, DATA)$

例 外：无

### 3.9.7 SH



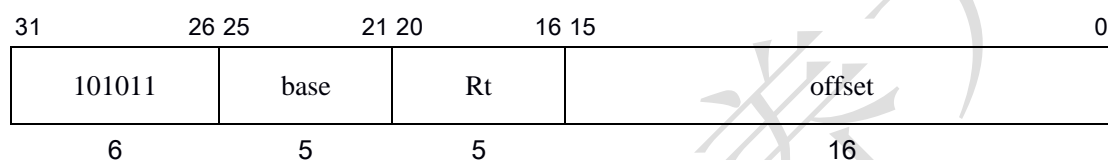
汇编格式: SH rt, offset(base)

功能描述: 将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址, 如果地址不是 2 的整数倍则触发地址错例外, 否则据此虚地址将 rt 寄存器的低半字存入存储器中。

操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
if  $vAddr_0 \neq 0$  then  
    SignalException(AddressError)  
endif  
(pAddr, CCA)  $\leftarrow$  AddressTranslation(vAddr, DATA, STORE)  
datahalf  $\leftarrow GPR[rt]_{15..0}$   
StoreMemory(CCA, HALFWORD, datahalf, pAddr, vAddr, DATA)

例 外: 地址最低 1 位不为 0, 触发地址错例外

### 3.9.8 SW



汇编格式: SW rt, offset(base)

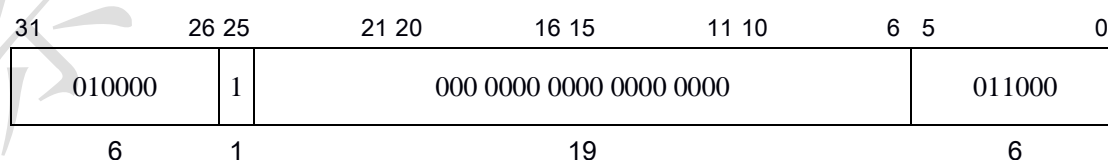
功能描述: 将 base 寄存器的值加上符号扩展后的立即数 offset 得到访存的虚地址, 如果地址不是 4 的整数倍则触发地址错例外, 否则据此虚地址将 rt 寄存器存入存储器中。

操作定义:  $vAddr \leftarrow GPR[base] + sign\_extend(offset)$   
if  $vAddr_{1..0} \neq 0$  then  
    SignalException(AddressError)  
endif  
(pAddr, CCA)  $\leftarrow$  AddressTranslation(vAddr, DATA, STORE)  
dataword  $\leftarrow GPR[rt]_{31..0}$   
StoreMemory(CCA, WORD, dataword, pAddr, vAddr, DATA)

例 外: 地址最低 2 位不为 0, 触发地址错例外

## 3.10 特权指令

### 3.10.1 ERET



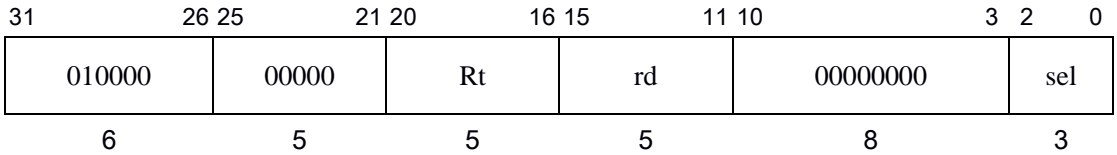
汇编格式: ERET

功能描述: 从中断、例外处理返回。

操作定义:  $PC \leftarrow EPC$

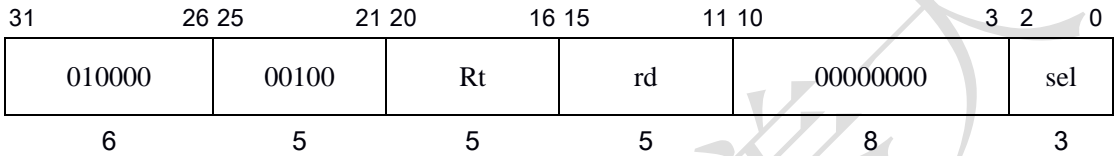
例 外: 无

3.10.2 MFC0



汇编格式: MFC0 rt, rd, sel  
功能描述: 从协处理器 0 的寄存器取值  
操作定义:  $GPR[rt] \leftarrow CP0[rd, sel]$   
例 外: 无

3.10.3 MTC0



汇编格式: MTC0 rt, rd, sel  
功能描述: 向协处理器 0 的寄存器存值  
操作定义:  $CP0[rd, sel] \leftarrow GPR[rt]$   
例 外: 无

4 存储管理

本指令系统对操作模式和存储访问类型进行了严格的限定，所以存储管理部分仅对虚实地址映射方式进行定义。

处理器可访问 32 位虚地址空间。整个 4GB 大小的虚地址空间被划分成 5 个段。处理器采用固定地址映射（FMT）机制，在进行虚实地址映射时，不同段的虚地址以线性方式固定地映射到一段连续的物理地址上。具体映射方式如图 4-1 所示。

---

## 参考文献

- [1] MIPS Architecture For Programmers Volume I-A: Introduction to the MIPS32 Architecture. Revision 3.02.
- [2] MIPS Architecture For Programmers Volume II-A: The MIPS32 Instruction Set. Revision 3.02.
- [3] MIPS Architecture For Programmers Vol. III: MIPS32/microMIPS32 Privileged Resource Architecture. Revision 3.02.
- [4] See MIPS Run Linux, 2nd Edition, 2006, Morgan Kaufmann, Dominic Sweetman