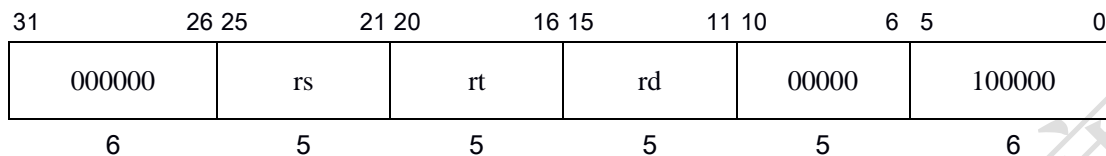


### 3.3 算术运算指令

#### 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:  $\text{tmp} \leftarrow (\text{GPR}[\text{rs}]_{31} \parallel \text{GPR}[\text{rs}]_{31..0}) + (\text{GPR}[\text{rt}]_{31} \parallel \text{GPR}[\text{rt}]_{31..0})$

if  $\text{tmp}_{32} \neq \text{tmp}_{31}$  then

    SignalException(IntegerOverflow)

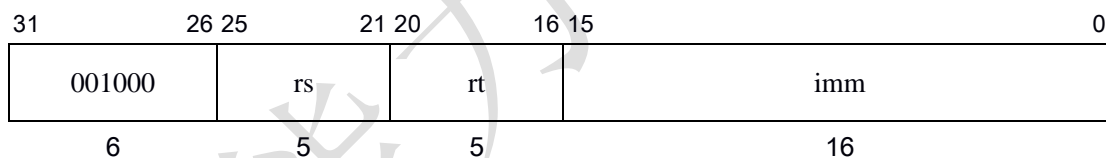
else

$\text{GPR}[\text{rd}] \leftarrow \text{tmp}_{31..0}$

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:  $\text{tmp} \leftarrow (\text{GPR}[\text{rs}]_{31} \parallel \text{GPR}[\text{rs}]_{31..0}) + \text{sign\_extend}(\text{imm})$

if  $\text{tmp}_{32} \neq \text{tmp}_{31}$  then

    SignalException(IntegerOverflow)

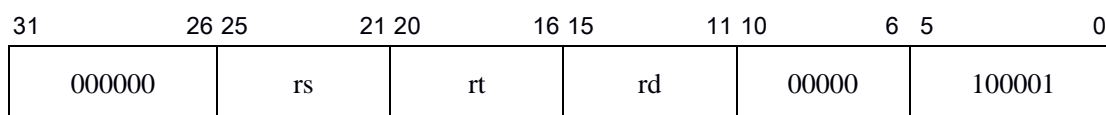
else

$\text{GPR}[\text{rt}] \leftarrow \text{tmp}_{31..0}$

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

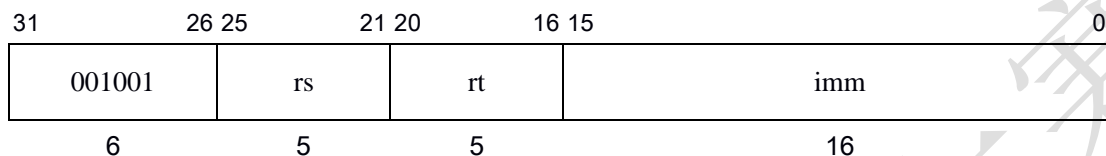
汇编格式: ADDU rd, rs, rt

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



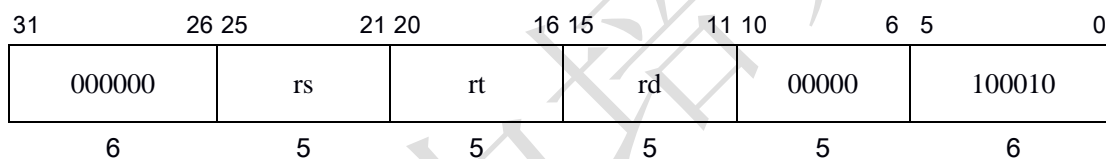
汇编格式: ADDIU rt, rs, imm

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

if  $\text{tmp}_{32} \neq \text{tmp}_{31}$  then

    SignalException(IntegerOverflow)

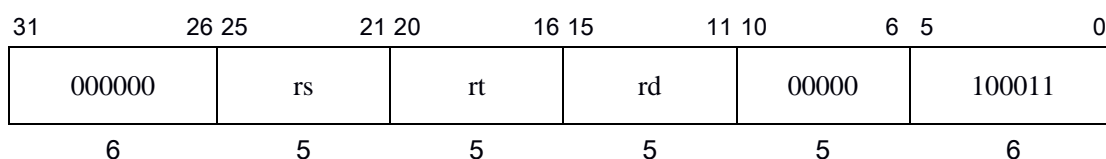
else

$GPR[rd] \leftarrow \text{tmp}_{31:0}$

endif

例 外: 如果有溢出, 则触发整型溢出例外。

### 3.3.6 SUBU



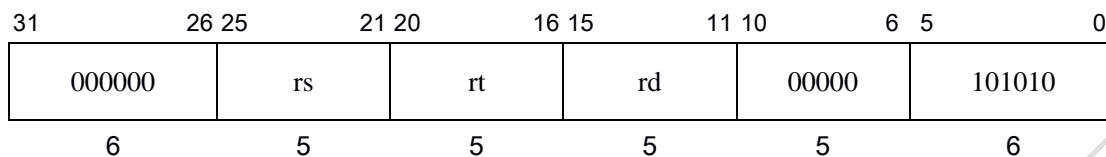
汇编格式: SUBU rd, rs, rt

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

$GPR[rd] \leftarrow 1$

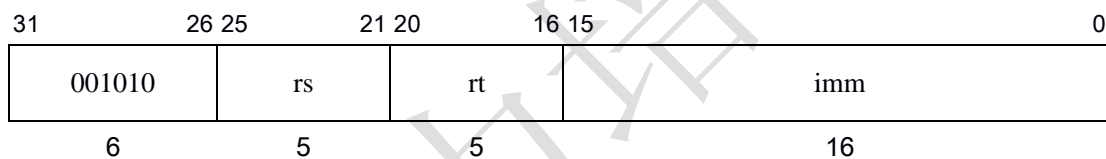
else

$GPR[rd] \leftarrow 0$

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

$GPR[rt] \leftarrow 1$

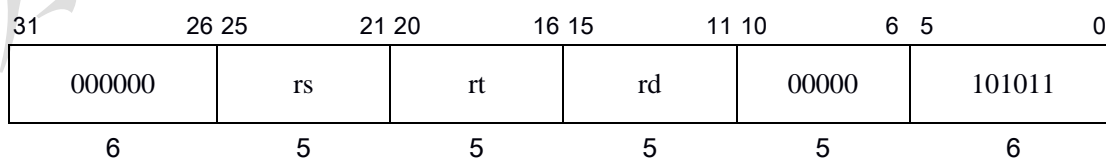
else

$GPR[rt] \leftarrow 0$

endif

例 外: 无

### 3.3.9 SLTU

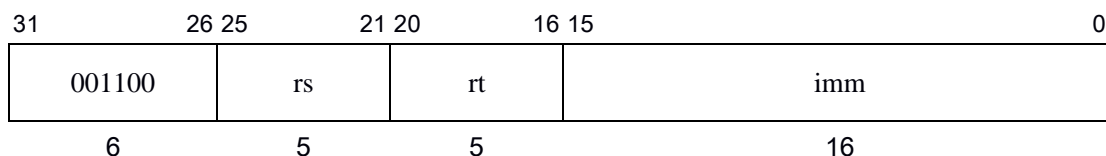


汇编格式: SLTU rd, rs, rt

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

### 3.4.2 ANDI



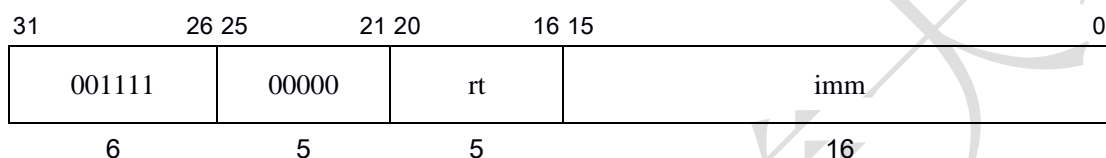
汇编格式: ANDI rt, rs, imm

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



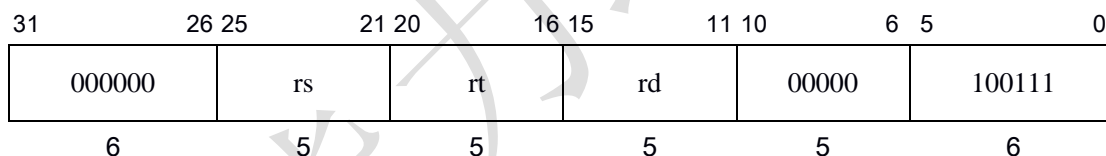
汇编格式: LUI rt, imm

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



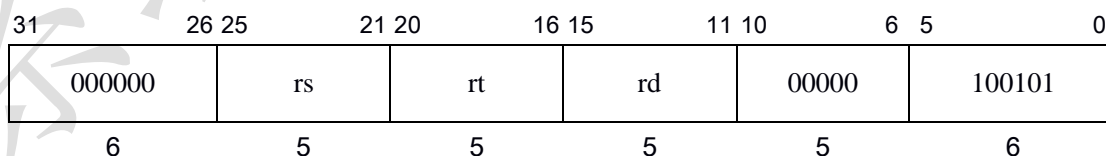
汇编格式: NOR rd, rs, rt

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



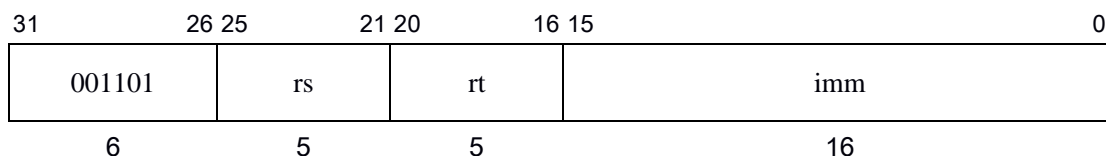
汇编格式: OR rd, rs, rt

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



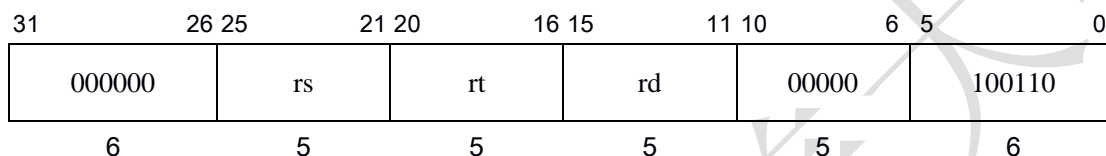
汇编格式: ORI rt, rs, imm

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



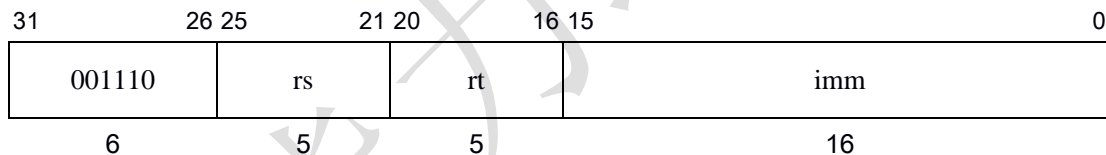
汇编格式: XOR rd, rs, rt

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



汇编格式: XORI rt, rs, imm

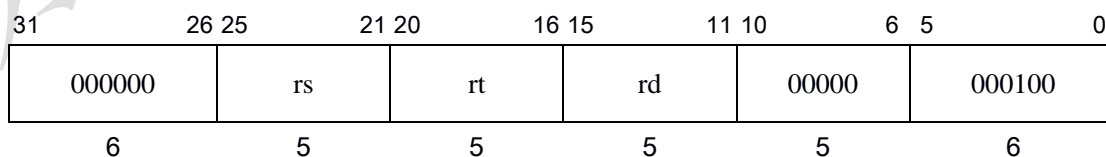
功能描述: 寄存器 rs 中的值与 0 扩展至 32 位的立即数 imm 按位逻辑异或, 结果写入寄存器 rt 中。

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



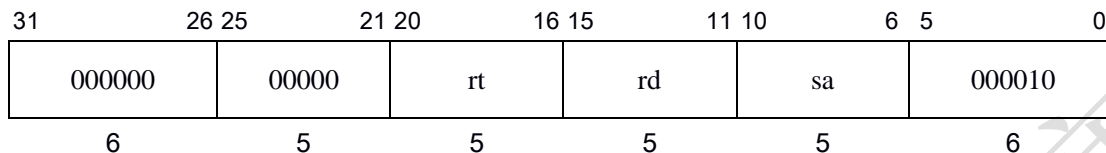
汇编格式: SLLV rd, rt, rs

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

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

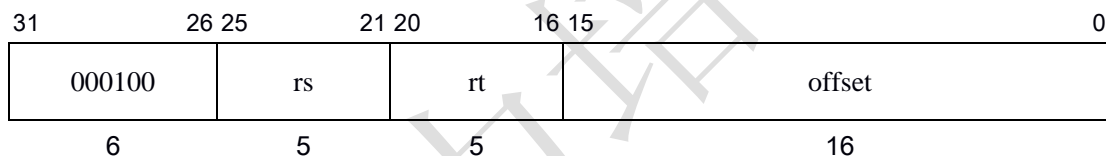
### 3.5.6 SRL



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

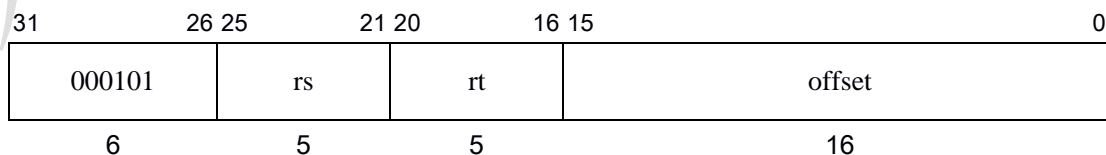
### 3.6.1 BEQ



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

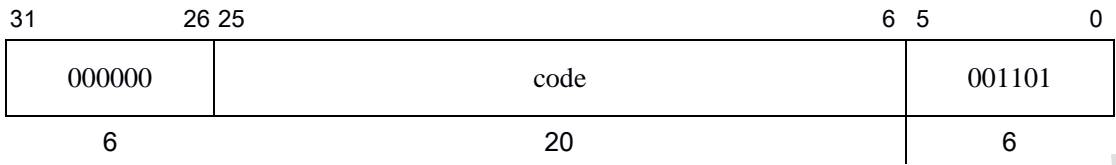
### 3.6.2 BNE



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。

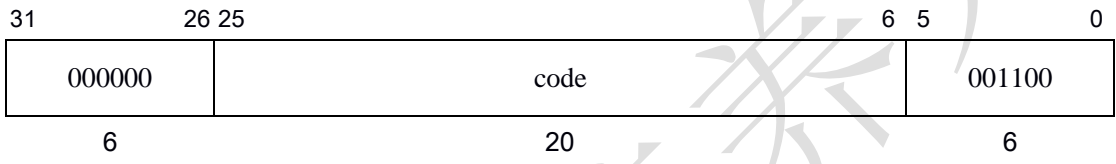
### 3.8 自陷指令

#### 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

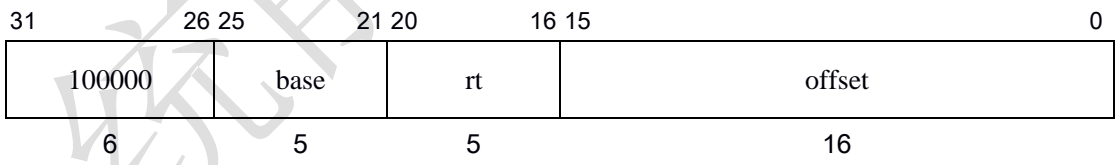
#### 3.8.2 SYSCALL



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

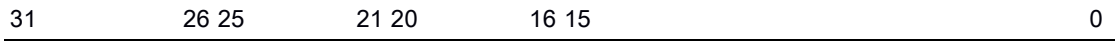
### 3.9 访存指令

#### 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



---

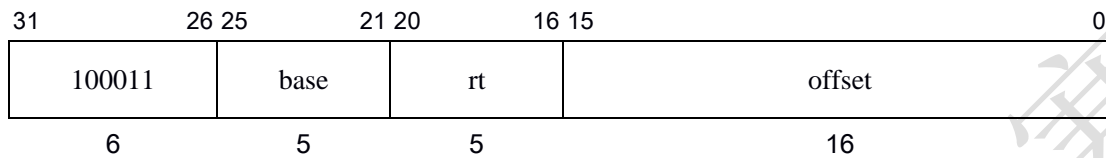
```

(pAddr, CCA)  $\leftarrow$  AddressTranslation(vAddr, DATA, LOAD)
memhalf  $\leftarrow$  LoadMemory(CCA, HALFWORD, pAddr, vAddr, DATA)
GPR[rt]  $\leftarrow$  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



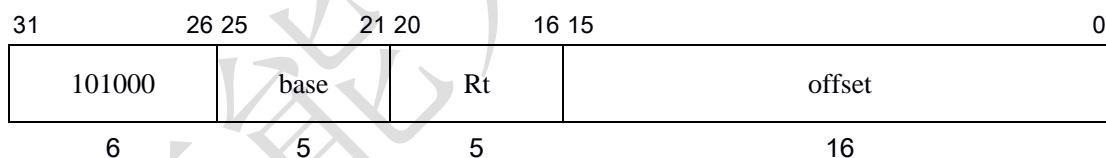
汇编格式：LW rt, offset(base)

**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



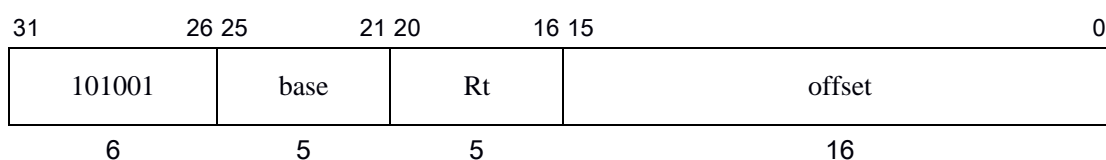
汇编格式：SB rt, offset(base)

**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



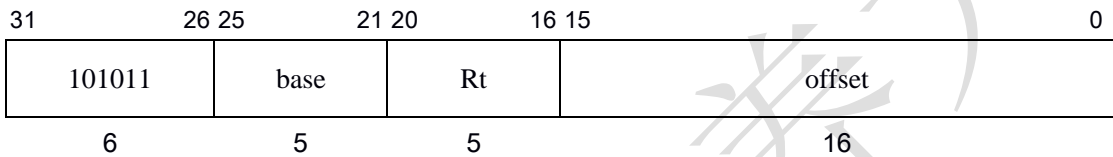
**汇编格式:** SH rt, offset(base)

**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