**R408指令手册**

**文档编号：20200517**

1. **基础指令**
   1. R-类指令

R类指令为通用寄存器-通用寄存器间的指令，指令编码为：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 15 | 14:12 | 11:9 | 8:6 | 5:2 | 1:0 |
| IMM[7] | RS2 | RS1 | RD | Funct4 | opcode |

R-Type指令的Opcode为2‘b00，Funct4编码段表示该指令：

|  |  |  |
| --- | --- | --- |
| Funct4 | 指令 | 解释 |
| 0000 | ADD | 加 |
| 0001 | SUB | 减 |
| 0010 | AND | 与 |
| 0011 | OR | 或 |
| 0100 | XOR | 异或 |
| 0101 | SR | 右移位，符号位拓展 |
| 0110 | SL | 左移位，移位量RS2 |
| 0111 | SRA | 右移位，无符号位拓展 |
| 1000 | SLT | RS1比较RS2，若RS1较小则写回1 |
| 1001 | SLTU | （无符号）RS1比较RS2，若RS1较小则写回1 |
| 1010 | EQ | RS1比较RS2，相等则写回1 |
| 1011 | NEQ | RS1比较RS2，不相等写回1 |

因为该指令编码中，立即数只能编码7位，故在R类指令的第15位加入符号位，在运算完成后，该位和运算结果的第7位做或运算写回RD寄存器。

* 1. B类指令

B类指令为分支类指令，用于分支指令。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15:12 | 11:9 | 8:6 | 5:2 | 1:0 |
| Imm[7:3] | RS1 | Imm[2:0] | Funct4 | opcode |

B-Type指令编码为2’b01，Funct4编码段表示该指令：

|  |  |  |
| --- | --- | --- |
| Funct4 | 指令 | 解释 |
| 0000 | BRA0 | 若源寄存器的bit0=1则分支 |

为了能更好的利用16位的编码空间，我决定在R-Type指令中添加了SLT，SLTIU，EQ，NEQ等指令来获取标志位。B-Type指令跳转地址为指令中编码的8位立即数左移2位，进行符号位拓展到16位之后和当前PC相加，这样可以跳转到前后512字节的空间。

如果要进行一个条件分支，可以进行如下操作：

*EQ R3, R4, R5 ；比较R4和R5，比较结果写回R3*

*BRA0 R3，#Lable ；若R3的bit0=1，则跳转*

* 1. SYS 类指令

SYS类指令为跳转-链接，寄存器-控制寄存器之间的操作，其指令编码为：

|  |  |  |  |
| --- | --- | --- | --- |
| 15:9 | 8:6 | 5:2 | 1:0 |
| CR | RD | Funct4 | opcode |

SYS类指令的Opcode为2’b10

|  |  |  |
| --- | --- | --- |
| Funct4 | 指令 | 解释 |
| 0000 | JL | 跳转链接，跳转到R6，R7指示的地址处，链接寄存器为CPC |
| 0001 | APC | 获取PC，写回CPC寄存器 |
| 0010 | JMP | 绝对跳转，跳转地址为R6和R7 |
| 0011 | WCR | 写控制寄存器，值为R6和R7 |
| 0100 | RCR | 读控制寄存器，值存回R0和R1 |
| 0101 | RET | 返回 |
| 1111 | FENCE | 存储器屏障 |

\*R408指令集设计目的是单周期实现的控制器，fence指令不是有很大必要

* 1. LS类指令

LS类指令为存取指令，包含载入立即数的操作，指令编码为：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15:12 | 11:9 | 8:6 | 5:2 | 1:0 |
| Imm[6:3] | Imm[2:0] or RS1 | RD | Funct4 | opcode |

LS类指令的Opcode为 2’b11，Funct4段表示该指令：

|  |  |  |
| --- | --- | --- |
| Funct4 | 指令 | 解释 |
| 0000 | LI | 载入7位立即数（高位补符号位）到RD寄存器 |
| 1000 | LB | 从内存中读取一个值到RD |
| 1001 | SB | 存RD寄存器,地址为R6和R7指定 |

如果要载入一个完整的8位立即数，可以如下操作：

*LI R1, 0x5*

*OR R1, R1, R1, 0x1*

这样就可以得到一个完整的8位立即数。

特别的！如果要访问内存，请连续使用两条相同的访问内存指令。

1. **CR寄存器**

CR寄存器是R408架构中的控制寄存器，一共定义了128个16位CR，目前使用的CR有：

|  |  |  |
| --- | --- | --- |
| 地址 | 名称 | 功能 |
| 0x00 | STATU | 机器控制 |
| 0x01 | IE | 中断使能 |
| 0x02 | EPC | 异常时的PC |
| 0x03 | CPC | 调用链接PC |
| 0x04 | TEMP | 临时值，作为一个临时寄存器可以保存一些值 |
| 0x05 | TVEC0 | 中断时候的跳转地址，中断0（最高优先级） |
| 0x06 | TVEC1 | 中断时候的跳转地址，中断1 |
| 0x07 | TVEC2 | 中断时候的跳转地址，中断2 |
| 0x08 | TVEC3 | 中断时候的跳转地址，中断3 |
| 0x09+ | 保留 | 保留 |

2.1、STATU寄存器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15:4 | 3 | 2 | 1 | 0 |
| 保留 | 保留 | 保留 | PGIE | GIE |

2.2、IE寄存器

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 15：4 | 3 | 2 | 1 | 0 |
| 保留 | IE3 | IE2 | IE1 | IE0 |