

R408 指令手册

文档编号：20200517

1、 基础指令

1.1、 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。

1.2、 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
```

1.3、 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

1.4、 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。

2、 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