MIPS-C指令测试指导文档

1. 指令功能测试+部分转发测试

CALR

add:

1. 正+正：0x10a38180, 0x40bf1920
2. 正+负：0x10a38180, 0x8abd9108
3. 负+正：0x8abd9108，0x10a38180
4. 负+负：0x8abd9108，0xffffffff

addu：

1. 正+正：0x10a38180, 0x40bf1920
2. 正+负：0x10a38180, 0x8abd9108
3. 负+正：0x8abd9108，0x10a38180
4. 负+负：0x8abd9108，（负+正）
5. 正正溢出：0x7192f91f 0x71029389
6. 负负溢出：0x8abd9108, 0x8abd9108

sub：

1. 正-正：100-101
2. 正-负：32-（-109213）
3. 负-负：-16 - （-1）
4. 负-正：-16 – 1

subu：

1. 正-正：100-101
2. 正-负：32-（-109213）
3. 负-负：-100 - （-1）
4. 负-正：-100 – 1
5. 正-负（溢出）：0x7102a81b – 0x81a0293
6. 负-正（溢出）：0x80000000 - 0x7fffffff

sll：

1. 左移0位：0x12345678 << 0
2. 左移2位：0x12345678 << 2
3. 左移24位：0x12345678 << 24

srl:

1. 右移0位：0x87654321 >> 0
2. 右移2位：0x87654321 >> 2
3. 右移24位：0x87654321 >> 24

sra:

1. 右移4位：0x87654321 >> 4
2. 右移32位：0x87654321 >> 31
3. 右移6位：0x12345678 >> 6

sllv:

1. 左移0位：0x12345678 << 0
2. 左移2位：0x12345678 << 2
3. 左移24位：0x12345678 << 24
4. 左移48位（溢出）：0x12345678 << 48

srlv:

1. 右移0位：0x87654321 >> 0
2. 右移2位：0x87654321 >> 2
3. 右移24位：0x87654321 >> 24
4. 右移48位（溢出）：0x87654321 >> 48

srav：

1. 右移4位：0x87654321 >> 4
2. 右移32位：0x87654321 >> 31
3. 右移16位（溢出）：0x87654321 >> 48
4. 右移6位：0x12345678 >> 6

and:

1. 全相交：0xffff0101,0xffff0101
2. 全不相交：0x77773333,0x8888cccc
3. 部分相交：111 101

or：

1. 相反：0x77773333,0x8888cccc
2. 部分：011 101
3. 0 0xabc77777

xor：

1. 相反：0x77773333,0x8888cccc
2. 相等：0xffff0101, 0xffff0101
3. 部分：0xfabc782f 0xd728cbaa

nor:

1. 相反：0x77773333,0x8888cccc
2. 部分：011 101
3. 0 0xabc77777

slt:

1. 负正：0xfffffff0, 1092
2. 正负：1092，-123
3. 负负小：0xfffffff0，-1
4. 负负大：-1，0xfffffff0
5. 正正小：1092，0x000f0000
6. 正正大：1092,0

sltu：

1. 负正：0xfffffff0, 1092
2. 正负：1092，0xfffffff0
3. 负负小：0xfffffff0，-1
4. 负负大：-1，0xfffffff0
5. 正正小：1092，0x000f0000
6. 正正大：1092,0

HILO寄存器操作类：测试方式-乘除->读寄存器->交换HILO寄存器值->再读

mult

1. 0x71923091, 0
2. 0x71923091, 0x61098239
3. -193, 100
4. 100, -193
5. -193, -193

multu:

1. 0x71923091, 0
2. 0x71923091, 0x61098239
3. -193, 100
4. 100, -193
5. -193, -193

div：

1. 100, 11
2. -100, -3
3. -100, 17
4. 100, -17

divu：

1. 100, 11
2. -100, -3
3. -100, 17
4. 100, -17

CALI：

addi:

1. 0+正：0+2018
2. 0+负：0+（-2018）
3. 正+负：2018+（-2018）
4. 负+正：（-2018）+2018
5. 正正：2018+2018
6. 负负：-2018+（-2018）

addiu：

1. 0+正：0+2018
2. 0+负：0+（-2018）
3. 正+负：2018+（-2018）
4. 负+正：（-2018）+2018
5. 正正：2018+2018
6. 负负：-2018+（-2018）
7. 正正（溢出）：0x7fffffff + 100
8. 负负（溢出）：0x800000ff + 0xfffff000

andi：

1. 0 & 0xffff
2. 探测扩展：0xffffffff & 0x8888

ori：

1. 探测扩展：0 | 0xffff
2. 0xffff7777 | 0x3333

xori:

1. 探测扩展：0 ^ 0xffff
2. 0xffff7777 ^ 0x3333

lui:

1. 正数：0x7001
2. 负数：0xffff

slti:

1. 正正小：100 10000
2. 正正大：10000 100
3. 负负小：-2 -1
4. 负负大：-1 -2
5. 负正：-1 0x7fff
6. 正负：0x7fff -1

sltiu:

1. 正正小：100 10000
2. 正正大：10000 100
3. 负负小：-2 -1
4. 负负大：-1 -2
5. 负正：-1 0x7fff
6. 正负：0x7fff -1

ST:

sb

1. 0x12345678 -> -1(512)
2. 0xaabbccdd -> -2(512)
3. 0x87654321 -> -3(512)
4. 0xddccbbaa -> -4(512)
5. 0x12345678 -> 1(512)
6. 0xaabbccdd -> 2(512)
7. 0x87654321 -> 3(512)
8. 0xddccbbaa -> 4(512)

sh

1. 0xaabbccdd -> -2(2048)
2. 0xddccbbaa -> -4(2048)
3. 0xaabbccdd -> 2(2048)
4. 0xddccbbaa -> 4(2048)

sw

1. 0x12345678 -> -8(1024)
2. 0xaabbccdd -> -4(1024)
3. 0x87654321 -> 0(1024)
4. 0xddccbbaa -> 4(1024)

LD:

lw:

1. -7(1023)
2. -3(1023)
3. 1(1023)
4. 5(1023)

lh:

1. 0(1024)
2. 2(1024)
3. -2(1024)
4. -4(1024)

lhu:

1. 0(1024)
2. 2(1024)
3. -2(1024)
4. -4(1024)

lb:

1. 0(1024)
2. 1(1024)
3. 2(1024)
4. 3(1024)
5. -1(1024)
6. -2(1024)
7. -3(1024)
8. -4(1024)

lbu:

1. 0(1024)
2. 1(1024)
3. 2(1024)
4. 3(1024)
5. -1(1024)
6. -2(1024)
7. -3(1024)
8. -4(1024)

BRANCH:

指导原则：

1. 正跳、负跳、正不跳、负不跳
2. 延迟槽探测
3. 操作数：0,1,2,-1,-2
4. 相等：正正负负
5. 不相等：正正、负负、正负

J\_Type：

指导原则：

1. 延迟槽检测
2. 函数样式调用测试
3. 递归调用综合程序检查（最后一步）。
4. jal和jalr赋值下的转发检测。