

New issue



Incorrect decode mask for vsetvli/vsetivli/vsetvl #952

Closed #958

Assignees

youzi27 opened on Nov 29, 2025

Describe the bug

NEMU decodes `vsetvli`, `vsetivli`, `vsetvl` incorrectly. According to the RISC-V Vector spec, these instructions require `funct3 = 111` (bits 14–12). However, NEMU uses `???` for `funct3`, causing invalid instructions to be decoded as `vset*`.

To Reproduce

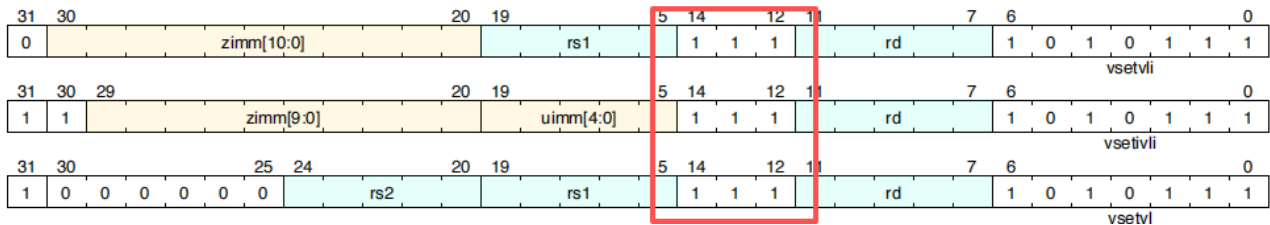
Check the current decode table:

```
def_INSTR_TAB("0?????? ???? ???? ? ???? 1010111", vsetvli);
def_INSTR_TAB("11???? ???? ???? ? ???? 1010111", vsetivli);
def_INSTR_TAB("1000000 ???? ???? ? ???? 1010111", vsetvl);
```

Any instruction with opcode 1010111 will match regardless of `funct3`.

Expected behavior

Formats for Vector Configuration Instructions under OP-V major opcode



Error log or Screenshots

```
[27] exception pc 0000000080001018 inst 0edd6357 cause 0000000000000002 <--
===== REF Regs =====
```

----- Intger Registers -----

\$0: 0x0000000000000000 ra: 0x0000000000000000 sp: 0x0000000000000000 gp:
0x0000000000000000
tp: 0x0000000000000000 t0: 0x0000000080000000 t1: 0x0000000000000000 t2:
0x0000000000000000
s0: 0x0000000000000000 s1: 0x0000000000000000 a0: 0x0000000000000000 a1:
0x0000000000000000
a2: 0x0000000000000000 a3: 0x0000000080000062 a4: 0x0000000000000000 a5:
0x0000000000000000
a6: 0x000000000000000f a7: 0x0000000000000000 s2: 0x0000000000000000 s3:
0x0000000000000000
s4: 0x0000000000000000 s5: 0x0000000000000000 s6: 0x0000000000000000 s7:
0x0000000000000000
s8: 0x0000000000000000 s9: 0x0000000000000000 s10: 0x0000000000000000 s11:
0x0000000000000000
t3: 0x0000000000000000 t4: 0x0000000000000000 t5: 0x0000000000000000 t6:
0x0000000000000000

----- Float Registers -----

ft0: 0xa9bd54c581dd5345 ft1: 0x135d9b10394d564a ft2: 0x38822fb5923bc130 ft3:
0x8712b2c7b5fde30f
ft4: 0xfb5746c912f81ad2 ft5: 0xde5ef1506d52847b ft6: 0x4e97bceaab0e07df ft7:
0xf06c11d049c8b026
fs0: 0xa340ed8eab0d9599 fs1: 0xdf458e543c971798 fa0: 0xec12c00870c78d25 fa1:
0x0d1589c64ce727b3
fa2: 0x4e8990575cc29e62 fa3: 0x04fd30a01c062ec4 fa4: 0x179c6a13f17f5e99 fa5:
0x63b470aaf563ca83
fa6: 0xe422dfafbef1eddb fa7: 0xc385a22ff91c8b44 fs2: 0x68d975b7a018626f fs3:
0x56153110e043a661
fs4: 0x56e26e5d43ee10d3 fs5: 0x0000000000000000 fs6: 0xde6a624158451ce4 fs7:
0x0b87666bdb329239
fs8: 0xc228c163d1bcb313 fs9: 0x530dcddb03d80058 fs10: 0xea92f75ade8099bb fs11:
0xbef9b45da40171dd
ft8: 0x5099a5df4028e333 ft9: 0x418740071a07d6fc ft10: 0x73ae5b72d5414904 ft11:
0x5b39a09bd0077be5
fcsr: 0x0000000000000000 fflags: 0x0000000000000000 frm: 0x0000000000000000

----- Privileged CSRs -----

pc: 0x000000008000101c privilege mode: U (mode: 0 v: 0 debug: 0)
mstatus: 0x8000004a00446788 sstatus: 0x800000200046700 vsstatus: 0x0000000200000000
hstatus: 0x0000000200000000 mnstatus: 0x0000000000000008
mcause: 0x0000000000000000 mepc: 0x0000000080000062 mtval: 0x0000000000000000
scause: 0x0000000000000000 sepc: 0x0000000000000000 stval: 0x0000000000000000
vscache: 0x0000000000000000 vsepc: 0x0000000000000000 vstval: 0x0000000000000000
mncause: 0x0000000000000000 mnepc: 0x0000000000000000 mnscratch: 0x0000000000000000
mtval2: 0x0000000000000000 htval: 0x0000000000000000
mtinst: 0x0000000000000000 htinst: 0x0000000000000000
mscratch: 0x04d607169a82bd3e sscratch: 0x5142142f6fd11beb vsscratch: 0x8f4461d3e016af80
mtvec: 0x0000000080001000 stvec: 0x0000000000000000 vstvec: 0x0000000000000000
mip: 0x0000000000000000 mie: 0x0000000000000000
mideleg: 0x0000000000001444 medeleg: 0x0000000000000000
hideleg: 0x0000000000000000 hedeleg: 0x0000000000000000
satp: 0x0000000000000000 hgatp: 0x0000000000000000 vsatp: 0x0000000000000000
mcounteren: 0x0000000000000000 scounteren: 0x0000000000000000 hcounteren:
0x0000000000000000
miselect: 0x0000000000000000 siselect: 0x0000000000000000 vsiselect: 0x0000000000000000
mireg: 0x0000000000000000 sireg: 0x0000000000000000 vsireg: 0x0000000000000000
mtopi: 0x0000000000000000 stopi: 0x0000000000000000 vstopi: 0x0000000000000000

```

mvien: 0x0000000000000000    hvien: 0x0000000000000000    mvip: 0x0000000000000000
mtopei: 0x0000000000000000    stopei: 0x0000000000000000    vstopei: 0x0000000000000000
hvictl: 0x0000000000000000    hviprio1: 0x0000000000000000    hviprio2: 0x0000000000000000

```

----- PMP CSRs -----

pmp: 32 entries active, details:

```

0: cfg:0x0f addr:0x0000000080001010| 1: cfg:0x00 addr:0x0000000000000000
2: cfg:0x00 addr:0x0000000000000000| 3: cfg:0x00 addr:0x0000000000000000
4: cfg:0x00 addr:0x0000000000000000| 5: cfg:0x00 addr:0x0000000000000000
6: cfg:0x00 addr:0x0000000000000000| 7: cfg:0x00 addr:0x0000000000000000
8: cfg:0x00 addr:0x0000000000000000| 9: cfg:0x00 addr:0x0000000000000000
10: cfg:0x00 addr:0x0000000000000000|11: cfg:0x00 addr:0x0000000000000000
12: cfg:0x00 addr:0x0000000000000000|13: cfg:0x00 addr:0x0000000000000000
14: cfg:0x00 addr:0x0000000000000000|15: cfg:0x00 addr:0x0000000000000000
16: cfg:0x00 addr:0x0000000000000000|17: cfg:0x00 addr:0x0000000000000000
18: cfg:0x00 addr:0x0000000000000000|19: cfg:0x00 addr:0x0000000000000000
20: cfg:0x00 addr:0x0000000000000000|21: cfg:0x00 addr:0x0000000000000000
22: cfg:0x00 addr:0x0000000000000000|23: cfg:0x00 addr:0x0000000000000000
24: cfg:0x00 addr:0x0000000000000000|25: cfg:0x00 addr:0x0000000000000000
26: cfg:0x00 addr:0x0000000000000000|27: cfg:0x00 addr:0x0000000000000000
28: cfg:0x00 addr:0x0000000000000000|29: cfg:0x00 addr:0x0000000000000000
30: cfg:0x00 addr:0x0000000000000000|31: cfg:0x00 addr:0x0000000000000000
32: cfg:0x00 addr:0x0000000000000000|33: cfg:0x00 addr:0x0000000000000000
34: cfg:0x00 addr:0x0000000000000000|35: cfg:0x00 addr:0x0000000000000000
36: cfg:0x00 addr:0x0000000000000000|37: cfg:0x00 addr:0x0000000000000000
38: cfg:0x00 addr:0x0000000000000000|39: cfg:0x00 addr:0x0000000000000000
40: cfg:0x00 addr:0x0000000000000000|41: cfg:0x00 addr:0x0000000000000000
42: cfg:0x00 addr:0x0000000000000000|43: cfg:0x00 addr:0x0000000000000000
44: cfg:0x00 addr:0x0000000000000000|45: cfg:0x00 addr:0x0000000000000000
46: cfg:0x00 addr:0x0000000000000000|47: cfg:0x00 addr:0x0000000000000000
48: cfg:0x00 addr:0x0000000000000000|49: cfg:0x00 addr:0x0000000000000000
50: cfg:0x00 addr:0x0000000000000000|51: cfg:0x00 addr:0x0000000000000000
52: cfg:0x00 addr:0x0000000000000000|53: cfg:0x00 addr:0x0000000000000000
54: cfg:0x00 addr:0x0000000000000000|55: cfg:0x00 addr:0x0000000000000000
56: cfg:0x00 addr:0x0000000000000000|57: cfg:0x00 addr:0x0000000000000000
58: cfg:0x00 addr:0x0000000000000000|59: cfg:0x00 addr:0x0000000000000000
60: cfg:0x00 addr:0x0000000000000000|61: cfg:0x00 addr:0x0000000000000000
62: cfg:0x00 addr:0x0000000000000000|63: cfg:0x00 addr:0x0000000000000000

```

----- PMA CSRs -----

pma: 32 entries active, details:

```

0: cfg:0x00 addr:0x0000000000000000| 1: cfg:0x00 addr:0x0000000000000000
2: cfg:0x00 addr:0x0000000000000000| 3: cfg:0x00 addr:0x0000000000000000
4: cfg:0x00 addr:0x0000000000000000| 5: cfg:0x00 addr:0x0000000000000000
6: cfg:0x00 addr:0x0000000000000000| 7: cfg:0x00 addr:0x0000000000000000
8: cfg:0x00 addr:0x0000000000000000| 9: cfg:0x00 addr:0x0000000000000000
10: cfg:0x00 addr:0x0000000000000000|11: cfg:0x00 addr:0x0000000000000000
12: cfg:0x00 addr:0x0000000000000000|13: cfg:0x00 addr:0x0000000000000000
14: cfg:0x00 addr:0x0000000000000000|15: cfg:0x00 addr:0x0000000000000000
16: cfg:0x00 addr:0x0000000000000000|17: cfg:0x00 addr:0x0000000000000000
18: cfg:0x00 addr:0x0000000000000000|19: cfg:0x0b addr:0x0000000040000000
20: cfg:0x0f addr:0x0000000080000000|21: cfg:0x0b addr:0x00000000c0040000
22: cfg:0x0b addr:0x00000000c0140000|23: cfg:0x0b addr:0x00000000e0080000
24: cfg:0x0f addr:0x00000000e0084000|25: cfg:0x0b addr:0x00000000e0088000
26: cfg:0x0b addr:0x00000000e4000000|27: cfg:0x0b addr:0x00000000e4008000
28: cfg:0x08 addr:0x00000000e8000000|29: cfg:0x0b addr:0x0000000020000000
30: cfg:0x6f addr:0x0000200000000000|31: cfg:0x18 addr:0x00001fffffffffffff

```

----- Vector Registers -----

```

v0 : 0xb3e3dc54c6817103_e5d52fc5cc96c091  v1 : 0x41042cb5b5f10328_8a0d049141f29f9c
v2 : 0x34a02554e893dd87_569cf02bff1c80ef  v3 : 0xc3e0ac0d5ffff42b0_c921d8f3b8a49222
v4 : 0x1455640c2c837ed2_1c12b594ebd7ddf6  v5 : 0xb502e24b353fa492_f7a39cfc3bbc4411
v6 : 0xa321721c8d5c27b2_594493007bee2bba  v7 : 0xbc401894584fd219_fedfcd9aad48a68b
v8 : 0xc37f4592dc1c2110_a91b4a6dbb07758c  v9 : 0x0d302c967b20d32f_0bca7664d0f6c3ad
v10: 0x7b6bc531cd1a4e21_29ef0b2a98b374e9  v11: 0x5c9450e692d00779_c8c76aca193e0197
v12: 0xd498f8547f04cf74_734d4b1c1fbeaa46  v13: 0x765db968a6545d1e_c0ef57e143bb32ad
v14: 0xe2481b4dda0abdea_4594d3c82841398d  v15: 0x16bee2ceb46ce309_91c1269ad299a578
v16: 0xc83eb71e25ce6fb8_7a49b759cfef39ce  v17: 0xf750eb6800c7bca1_c574beff845787a1
v18: 0x8b92c78998d3a348_a4e2e3cc906925f3  v19: 0xc55d40a30a67fabf_0da3a45024ce02a0
v20: 0x29230fff31a7502c6_e224a59197179c70  v21: 0x2756ab1931338f81_fb70b5bb7a4e60c7
v22: 0x7abdeeda99ae84bd_afc36c34584bb3a9  v23: 0xfeed2eece57a9229_a7357ddd18d2a5a7
v24: 0xc857b64919bc87bb_ea5be4c1dbc8ffa2  v25: 0x2887d57ee6160dcb_f4c1fabe2895dd9b
v26: 0x87c19af4a529206f_0cd43f5d61839947  v27: 0x7300ebdacf9562a0_a2a4d9b79dd34d82
v28: 0x5bcba38f1d82731_aaedbd7dbe953c84  v29: 0x78372ab9bc59689d_33cb7a6bf8e4a3e5
v30: 0x260bf930762bb89e_ffb640e85982b2a5  v31: 0xf48c2d0a2c660778_d41aa58285be6654
  vtype: 0x000000000000008e  vstart: 0x0000000000000000  vxsat: 0x0000000000000000
  vxrm: 0x0000000000000001   vl: 0x0000000000000000  vcsr: 0x0000000000000002

```

```
----- Triggers -----
```

```
tselect: 0x0000000000000000
```

```
0: tdata1: 0xf000000000000000 tdata2: 0x0000000000000000
```

```
1: tdata1: 0xf000000000000000 tdata2: 0x5f70696b735f7473
```

```
2: tdata1: 0xf000000000000000 tdata2: 0x722d6f742d796461
```

```
3: tdata1: 0xf000000000000000 tdata2: 0x2d756d656e2d3436
```

```
4: tdata1: 0x6572707265746e69 tdata2: 0x6f006f732d726574
```

```
privilegeMode: 3
```

```
mode different at pc = 0x0080001014, right= 0x0000000000000000, wrong =
0x0000000000000003
```

```
mstatus different at pc = 0x0080001014, right= 0x8000004a00446788, wrong =
0x8000040a00446780
```

```
mepc different at pc = 0x0080001014, right= 0x0000000800000062, wrong =
0x0000000080001018
```

```
mtval different at pc = 0x0080001014, right= 0x0000000000000000, wrong =
0x00000000edd6357
```

```
mcause different at pc = 0x0080001014, right= 0x0000000000000000, wrong =
0x0000000000000002
```

```
Core 0: ABORT at pc = 0xffff2c57116faab1
```

```
Core-0 instrCnt = 34, cycleCnt = 8,482, IPC = 0.004008
```

```
Seed=0 Guest cycle spent: 8,486 (this will be different from cycleCnt if emu loads a
snapshot)
```

```
Host time spent: 5,888ms
```

Necessary information on versions

- XiangShan commit: 10746d6aa982eb166f73cdb55c4334902ae604b9 (Wed Nov 26 17:46:13 2025)
- Ready-to-run(NEMU) commit: c4e0350c0f686cfa206d5b47d80cfd730f39675a (Fri Nov 7 18:17:19 2025)

Additional information

youzi27 mentioned this on [Nov 29, 2025](#)

[Fix incorrect vsetvi/vsetivli/vsetvl decode: funct3 must be 111 #953](#)

good-circle assigned [lewislzh](#) on [Dec 1, 2025](#)



cebarobot on Dec 1, 2025

Member



Hi, thanks for your issue.

Could you please provide a bin file to help use to reproduce this? I can't reproduce your issue.

My reproduction:

```

0000000080000000 <_start>:
    80000000: 00000297          auipc   t0,0x0
    80000004: 01828293          addi    t0,t0,24 # 80000018 <trap_handler>
    80000008: 30529073          csrw   mtvec,t0
    8000000c: 0edd6357          .word  0x0edd6357
    80000010: 4501              li     a0,0
    80000012: 0005006b          .word  0x0005006b

0000000080000016 <loop>:
    80000016: a001              j      80000016 <loop>

0000000080000018 <trap_handler>:
    80000018: 4501              li     a0,0
    8000001a: 0005006b          .word  0x0005006b

```



And got:

```

[ble: elapsed 2.612s (CPU 626.0%)] ./build/emu -i ../20251127-pma/test.bin --diff ./ready-to-run/riscv64-nemu-interpretor-so
11:05:38 fenghaoyuan@open102 20251015-perftest-1core ± || → ./build/emu -i ../20251127-pma/test.bin --diff ./ready-to-run/riscv64-nemu-interpretor-so 2>/dev/null
emu compiled at Oct 17 2025, 20:06:46
%Warning: System has stack size 8192 kb which may be too small; suggest 'ulimit -s 44930' or larger
Using simulated 32768B flash
Core 0's Commit SHA is: cb97d465b8, dirty: 0
Using simulated 8386560MB RAM
The image is ../20251127-pma/test.bin
DRAMsim3 config: /nfs/home/share/ci-workloads/DRAMsim3/configs/XiangShan.ini
DRAMsim3 outdir: /nfs/home/fenghaoyuan/20251015-perftest-1core/build
CPU_FREQ: 3000 DRAM_FREQ: 1600
CPU_CLOCK_CYCLE: 0.000333333 DRAM_CLOCK_CYCLE: 0.000625
DRAMsim3 memory system initialized.
The reference model is ./ready-to-run/riscv64-nemu-interpretor-so
The first instruction of core 0 has committed. Diffptest enabled.
Core 0: HIT GOOD TRAP at pc = 0x8000001a
Core-0 instrCnt = 7, cycleCnt = 4,331, IPC = 0.001616
Seed=0 Guest cycle spent: 4,336 (this will be different from cycleCnt if emu loads a snapshot)
Host time spent: 733ms
11:05:44 fenghaoyuan@open102 20251015-perftest-1core ± || → █

11:18:40 fenghaoyuan@open102 20251015-perftest-1core ± || → cat nemu-hart-0.log
[fetch_decode] (M) 0x0000000010000000: 9b 02 10 00      addiw   t0,$0,1
[fetch_decode] (M) 0x0000000010000004: 93 92 f2 01      c_slli  t0,t0,31
[fetch_decode] (M) 0x0000000010000008: 67 80 02 00      c_jr   $0,t0
[fetch_decode] (M) 0x0000000080000000: 97 02 00 00      auipc  t0,0x0
[fetch_decode] (M) 0x0000000080000004: 93 82 82 01      c_addi t0,t0,24
[fetch_decode] (M) 0x0000000080000008: 73 90 52 30      csrrw  $0,csr_0x305,t0
[fetch_decode] (M) 0x000000008000000c: 57 63 dd 0e      csrrw  $0,csr_0x305,t0
[fetch_decode] (M) 0x0000000080000018: 01 45              p_li_0 a0
11:18:47 fenghaoyuan@open102 20251015-perftest-1core ± || → █

```

The instruction trace shows that `0x0edd6357` at `0x8000000c` raises an exception and there is no diff.

By the way, the decode in NEMU is correct. NEMU decodes instructions by level. The `111` among bit 14-12 is decoded in `def_The1per(OP_V)` :

[NEMU/src/lisa/riscv64/instr/rvv/decode.h](#)

Lines 552 to 562 in `cfa9bc8`

```
552     def_The1per(OP_V) { // 10_101
553         def_INSTR_TAB("??????? ???? ???? 000 ???? ???? ??", vopivv);
554         def_INSTR_TAB("??????? ???? ???? 001 ???? ???? ??", vopfvv);
555         def_INSTR_TAB("??????? ???? ???? 010 ???? ???? ??", vopmvv);
556         def_INSTR_TAB("??????? ???? ???? 011 ???? ???? ??", vopivi);
557         def_INSTR_TAB("??????? ???? ???? 100 ???? ???? ??", vopivx);
558         def_INSTR_TAB("??????? ???? ???? 101 ???? ???? ??", vopfvf);
559         def_INSTR_TAB("??????? ???? ???? 110 ???? ???? ??", vopmvx);
560         def_INSTR_TAB("??????? ???? ???? 111 ???? ???? ??", vsetvl_dispatch);
561         return EXEC_ID_inv;
562     }
```

After `def_The1per(OP_V)` , the instruction will then be decoded by `def_The1per(vsetvl_dispatch)`:

[NEMU/src/lisa/riscv64/instr/rvv/decode.h](#)

Lines 544 to 549 in `cfa9bc8`

```
544     def_The1per(vsetvl_dispatch) {
545         def_INSTR_TAB("0?????? ???? ???? ??? ???? 1010111", vsetvli);
546         def_INSTR_TAB("11????? ???? ???? ??? ???? 1010111", vsetivli);
547         def_INSTR_TAB("1000000 ???? ???? ??? ???? 1010111", vsetvl);
548         return EXEC_ID_inv;
549     }
```

So the decode should be fine.



youzi27 on Dec 1, 2025

Author ...

It seems that the V environment has not been initialized on your side. I will upload the image file for you to test. If you have any questions, feel free to let me know.

testcase: [seeds_4646.zip](#)



cebarobot added a commit that references this issue [on Dec 2, 2025](#)

fix(vector): remove incorrect decode for some insts ...

4ecc1da



cebarobot mentioned this [on Dec 2, 2025](#)

[fix\(vector\): remove incorrect decode for some insts #958](#)



cebarobot on Dec 2, 2025

Member



Hi. Thanks for your workload and I found the problem.

The instruction `0x0edd6357` is decoded wrongly as `vredxor` (`funct6 = 000011`), under `vopmvx` (`funct3 = OPMVX`). Actually, the `funct3` of `vredxor.vs` could only be `OPMVV`.

				funct6		
				000000	V	vredsum
				000001	V	vredand
				000010	V	vredor
				000011	V	vredxor
				000100	V	vredminu
				000101	V	vredmin
				000110	V	vredmaxu
				000111	V	vredmax

Integer			
funct3			
OPMVV	V		
OPMVX		X	

I fixed this issue in [#958](#). Please try the diff-test from its CI.

The problem is indeed related to decoding, but not related to `vsetvli` / `vsetivli` / `vsetvl`.



cebarobot closed this as completed in [#958](#) on Dec 3, 2025



youzi27 on Dec 3, 2025

Author



Thank you for your detailed feedback and the fix.



cebarobot added a commit that references this issue on Dec 3, 2025

`fix(vector): remove incorrect decode for some insts` ([OpenXiangShan#958](#))



Verified

481de63

[Sign up for free](#) to join this conversation on GitHub. Already have an account? [Sign in to comment](#)

Metadata

Assignees



lewislzh

Labels

No labels

Type

No type

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development



Fix incorrect vsetvi/vsetivli/vsetvl decode: funct3 must be 111

OpenXiangShan/NEMU



fix(vector): remove incorrect decode for some insts

OpenXiangShan/NEMU

v2025.12.r2

Participants



