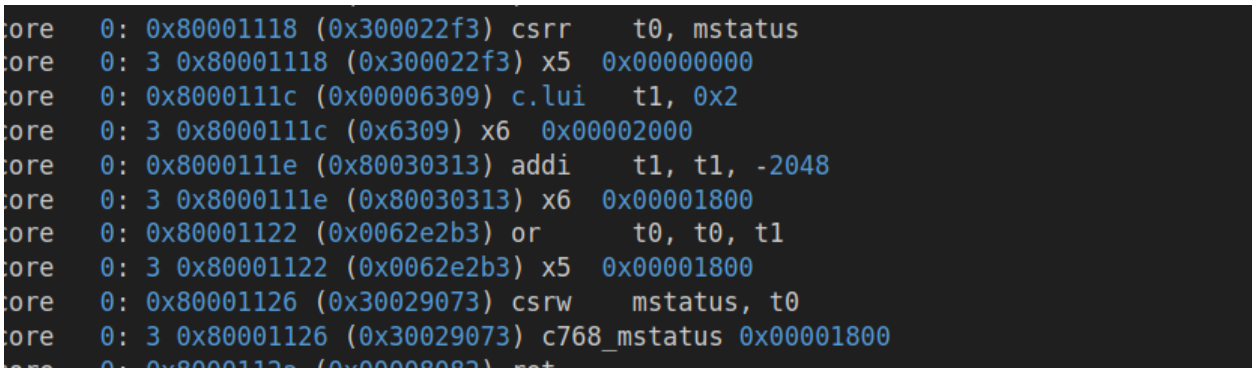
RISC-V arch test
Module: Privileged spec
Task 7

Test Description:

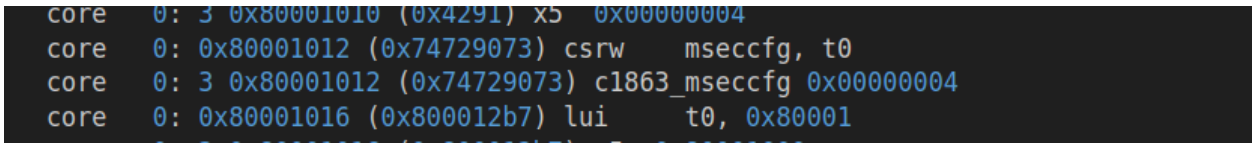
In this test I have verified the smepmp extension behavior. I have set the RLB (Rule locking bypass) bit to 1 which means bypass the locking rule. Now I set the pmp regions as we have to configure these regions with locked and permission bits and then made those regions as M-mone only regions. After that. After that I set the MML bit in mssecfg register to ensure that the rules are enforced on machine mode only.

After that I tried to change the permissions of the TOR region from read only to execute only and it did work because the mseccfg.RLB is set. If I don't set the mseccfg.RLB bit then permission won't get changed and pmpcfg writes are ignored.

Snapshots:



Start in M mode



Set the mseccfg.RLB bit so that we can change the permissions

```
core   0: 3 0x80001016 (0x800012b7) x5  0x80001000
core   0: 0x8000101a (0x0022d293) srli    t0, t0, 2
core   0: 3 0x8000101a (0x0022d293) x5  0x20000400
core   0: 0x8000101e (0x3b029073) csrw    pmpaddr0, t0
core   0: 3 0x8000101e (0x3b029073) c944_pmpaddr0 0x20000400
core   0: 0x80001022 (0x800012b7) lui     t0, 0x80001
core   0: 3 0x80001022 (0x800012b7) x5  0x80001000
core   0: 0x80001026 (0x0022d293) srli    t0, t0, 2
core   0: 3 0x80001026 (0x0022d293) x5  0x20000400
core   0: 0x8000102a (0x1ff2e293) ori     t0, t0, 511
core   0: 3 0x8000102a (0x1ff2e293) x5  0x200005ff
core   0: 0x8000102e (0x3b129073) csrw    pmpaddr1, t0
core   0: 3 0x8000102e (0x3b129073) c945_pmpaddr1 0x200005ff
core   0: 0x80001032 (0x0707a2b7) lui     t0, 0x707a
core   0: 3 0x80001032 (0x0707a2b7) x5  0x0707a000
core   0: 0x80001036 (0xc8928293) addi    t0, t0, -887
core   0: 3 0x80001036 (0xc8928293) x5  0x07079c89
core   0: 0x8000103a (0x3a029073) csrw    pmpcfg0, t0
core   0: 3 0x8000103a (0x3a029073) c928_pmpcfg0 0x07079c89
core   0: 0x8000103e (0x300022f3) csrr    t0, mstatus
```

Configure pmp regions

```
core   0: 3 0x80001042 (0x4285) x5  0x00000001
core   0: 0x80001044 (0x74729073) csrw    mseccfg, t0
core   0: 3 0x80001044 (0x74729073) c1863_mseccfg 0x00000001
core   0: 0x80001048 (0x0707a2b7) lui     t0, 0x707a
core   0: 3 0x80001048 (0x0707a2b7) x5  0x0707a000
core   0: 0x8000104c (0xc8c28293) addi    t0, t0, -884
```

Set the MML bit to enforce permissions only in M-mode

```
core   0: 3 0x80001048 (0x0707a2b7) x5  0x0707a000
core   0: 0x8000104c (0xc8c28293) addi    t0, t0, -884
core   0: 3 0x8000104c (0xc8c28293) x5  0x07079c8c
core   0: 0x80001050 (0x3a029073) csrw    pmpcfg0, t0
core   0: 3 0x80001050 (0x3a029073) c928_pmpcfg0 0x07079c89
core   0: 0x80001054 (0x00004285) c.li    t0, 1
core   0: 3 0x80001054 (0x4285) x5  0x00000001
```

Update the permissions to executable

Add a rule with executable privileges that either is M-mode-only or a locked Shared-Region?? what happens and why

Adding a rule with executable privileges that either is M-mode-only or a locked Shared-Region is not possible and such pmpcfg writes are ignored, leaving pmpcfg Unchanged.

How can you solve the above problem

The restrictions can be ignored by setting the mseccfg.RLB bit to 1. In this way the pmpcfg register writes does not get ignored. And pmpcfg writes to the new value.

References:

XLEN-1	10	9	8	7	3	2	1	0
WPRI		SSEED	USEED	**WPRI**		RLB	MMWP	MML
XLEN-10		1	1	5		1	1	1

Figure 3.26: Machine security configuration register (`mseccfg`).

Mseccfg register bits

3. On `mseccfg` we introduce a field in bit 1 called **Machine Mode Whitelist Policy (mseccfg.MMWP)**. This is a sticky bit, meaning that once set it cannot be unset until a **PMP reset**. When set it changes the default PMP policy for M-mode when accessing memory regions that don't have a matching PMP rule, to **denied** instead of **ignored**.
4. On `mseccfg` we introduce a field in bit 0 called **Machine Mode Lockdown (mseccfg.MML)**. This is a sticky bit, meaning that once set it cannot be unset until a **PMP reset**. When `mseccfg.MML` is set the system's behavior changes in the following way:
   a. The meaning of `pmpcfg.L` changes: Instead of marking a rule as **locked** and **enforced** in all modes, it now marks a rule as **M-mode-only** when set and **S/U-mode-only** when unset. The formerly reserved encoding of `pmpcfg.RW=01`, and the encoding `pmpcfg.LRWX=1111`, now encode a **Shared-Region**.

mseccfg.MML bit description