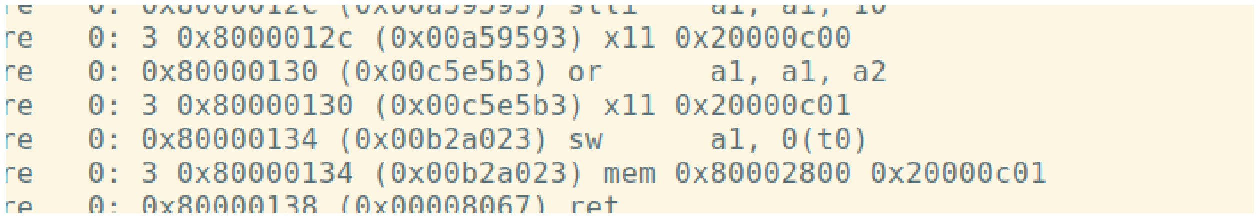
RV DV **Abdullah**

RISC V ARCH TEST
TASK 05

Test Description : First, set up the Level 1 and Level 0 page table entries.Then, write some instructions in the Data_and_int_page section. After that, test for instruction ,load, and store page faults from that section,which is mapped through MMU.

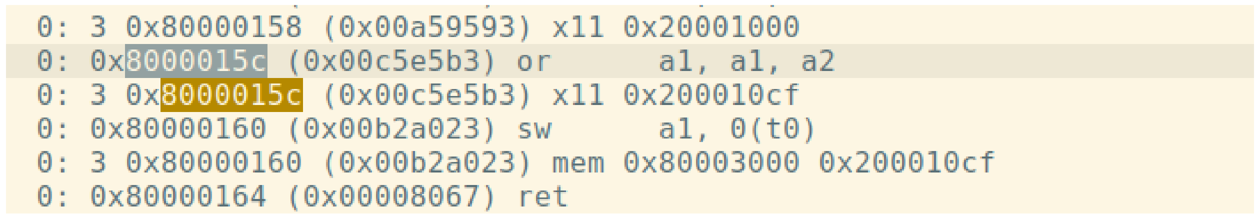
Virtual address space start form 0x80000000

Set root page PTE



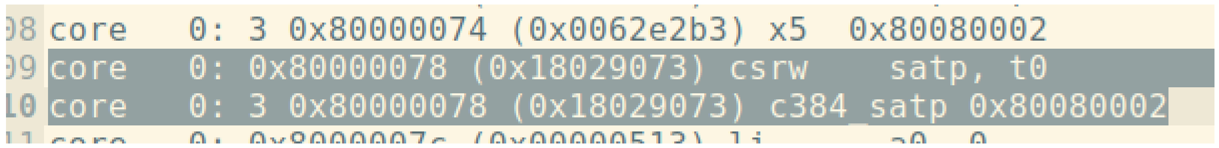
```
re   0: 3 0x8000012c (0x00a59593) x11 0x20000c00
re   0: 0x80000130 (0x00c5e5b3) or      a1, a1, a2
re   0: 3 0x80000130 (0x00c5e5b3) x11 0x20000c01
re   0: 0x80000134 (0x00b2a023) sw      a1, 0(t0)
re   0: 3 0x80000134 (0x00b2a023) mem 0x80002800 0x20000c01
```

Set leaf page entry with XWR and Access bit an dirty high



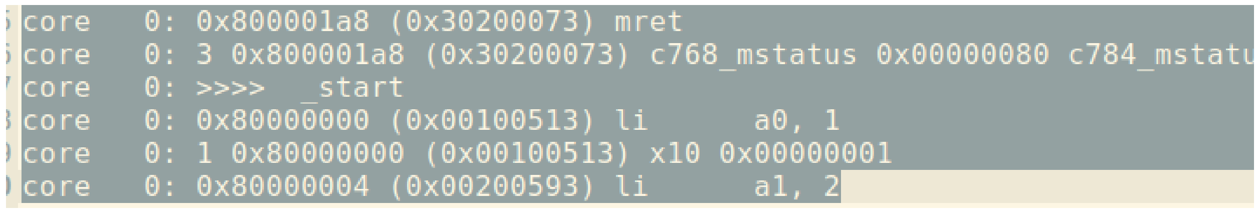
```
0: 3 0x80000158 (0x00a59593) x11 0x20001000
0: 0x8000015c (0x00c5e5b3) or      a1, a1, a2
0: 3 0x8000015c (0x00c5e5b3) x11 0x200010cf
0: 0x80000160 (0x00b2a023) sw      a1, 0(t0)
0: 3 0x80000160 (0x00b2a023) mem 0x80003000 0x200010cf
0: 0x80000164 (0x00008067) ret
```

Setup satap register with mode and PPN



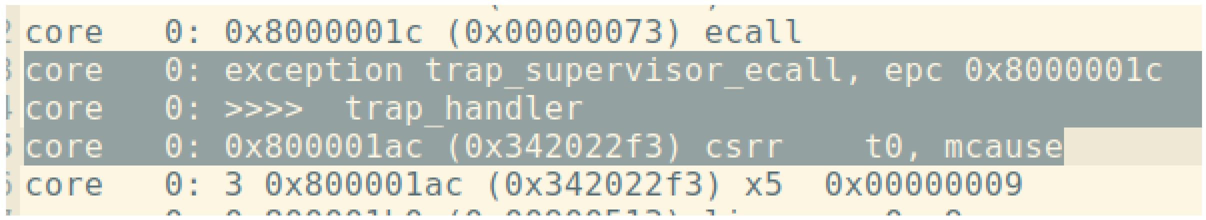
```
core   0: 3 0x80000074 (0x0062e2b3) x5  0x80080002
core   0: 0x80000078 (0x18029073) csrw    satp, t0
core   0: 3 0x80000078 (0x18029073) c384_satp 0x80080002
```

Switch to Supervisore mode and make sure mpec hold virtual address



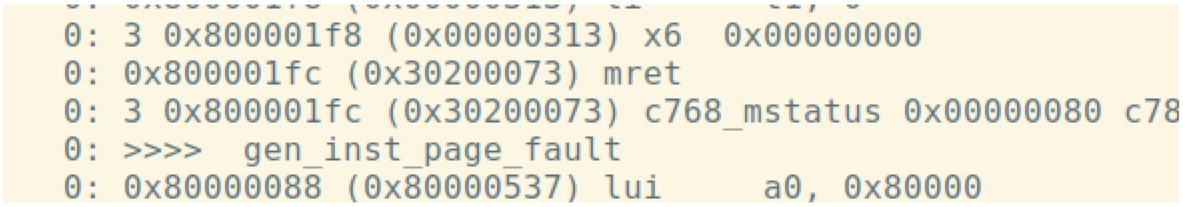
```
core   0: 0x800001a8 (0x30200073) mret
core   0: 3 0x800001a8 (0x30200073) c768_mstatus 0x00000080 c784_mstatu
core   0: >>>>  _start
core   0: 0x80000000 (0x00100513) li      a0, 1
core   0: 1 0x80000000 (0x00100513) x10 0x00000001
core   0: 0x80000004 (0x00200593) li      a1, 2
```

NOw call Ecall which switch supervisor mode to machine exception traphandler



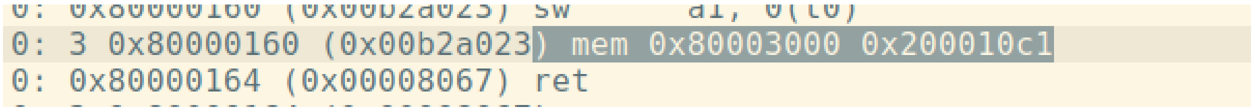
```
core   0: 0x8000001c (0x00000073) ecall
core   0: exception trap_supervisor_ecall, epc 0x8000001c
core   0: >>>>  trap_handler
core   0: 0x800001ac (0x342022f3) csrr    t0, mcause
core   0: 3 0x800001ac (0x342022f3) x5  0x00000009
```

Trap handler switch the and jump to main code linst_page_fault and change the level 0 PTE entry permissions ,



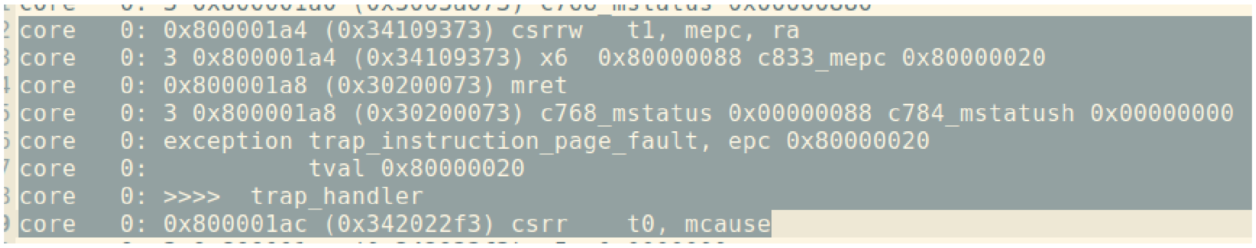
```
0: 3 0x800001f8 (0x00000313) x6  0x00000000
0: 0x800001fc (0x30200073) mret
0: 3 0x800001fc (0x30200073) c768_mstatus 0x00000080 c78
0: >>>>  gen_inst_page_fault
0: 0x80000088 (0x80000537) lui     a0, 0x80000
```

PTE permission becomes only READ WRITe WHICH cause the inst page fault



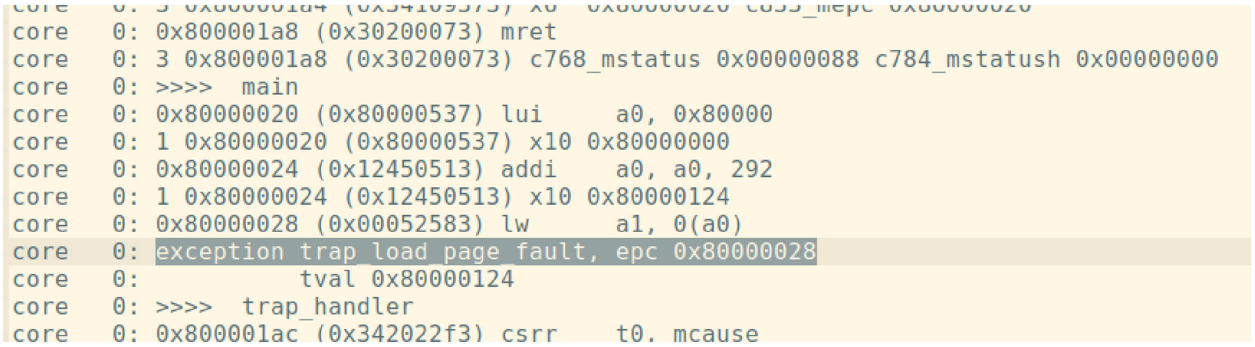
```
0: 3 0x80000160 (0x00b2a023) mem 0x80003000 0x200010c1
0: 0x80000164 (0x00008067) ret
```

Switch to supervisor mode make sure mepc loaded with virtual address and this cause the inst_page_fault



```
core   0: 0x800001a4 (0x34109373) csrrw   t1, mepc, ra
core   0: 3 0x800001a4 (0x34109373) x6  0x80000088 c833_mepc 0x80000020
core   0: 0x800001a8 (0x30200073) mret
core   0: 3 0x800001a8 (0x30200073) c768_mstatus 0x00000088 c784_mstatush 0x00000000
core   0: exception trap_instruction_page_fault, epc 0x80000020
core   0:           tval 0x80000020
core   0: >>>>  trap_handler
core   0: 0x800001ac (0x342022f3) csrr    t0, mcause
```

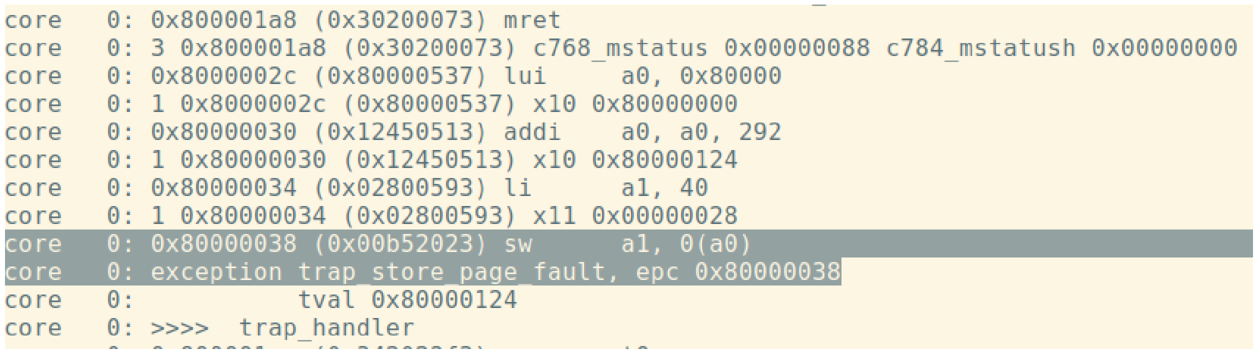
Then trape handler jump to main code according to mcause after that level 0 PTE permission updated to test load page fault and



```
core   0: 0x800001a8 (0x30200073) mret
core   0: 3 0x800001a8 (0x30200073) c768_mstatus 0x00000088 c784_mstatush 0x00000000
core   0: >>>>  main
core   0: 0x80000020 (0x80000537) lui     a0, 0x80000
core   0: 1 0x80000020 (0x80000537) x10 0x80000000
core   0: 0x80000024 (0x12450513) addi    a0, a0, 292
core   0: 1 0x80000024 (0x12450513) x10 0x80000124
core   0: 0x80000028 (0x00052583) lw      a1, 0(a0)
core   0: exception trap_load_page_fault, epc 0x80000028
core   0:           tval 0x80000124
core   0: >>>>  trap_handler
core   0: 0x800001ac (0x342022f3) csrr    t0, mcause
```

RISC V ARCH TEST
TASK 05

Now trape handler perform same operation but now set persian for store page fault



```
core   0: 0x800001a8 (0x30200073) mret
core   0: 3 0x800001a8 (0x30200073) c768_mstatus 0x00000088 c784_mstatush 0x00000000
core   0: 0x8000002c (0x80000537) lui     a0, 0x80000
core   0: 1 0x8000002c (0x80000537) x10 0x80000000
core   0: 0x80000030 (0x12450513) addi    a0, a0, 292
core   0: 1 0x80000030 (0x12450513) x10 0x80000124
core   0: 0x80000034 (0x02800593) li      a1, 40
core   0: 1 0x80000034 (0x02800593) x11 0x00000028
core   0: 0x80000038 (0x00b52023) sw      a1, 0(a0)
core   0: exception trap_store_page_fault, epc 0x80000038
core   0:           tval 0x80000124
core   0: >>>>  trap_handler
```

As we all tested write ,read exceted faults generated by permission now trape handler jump to main then it jump to test_pass , Hart is in m mode at time of exit