IOPMP Task Group Meeting

August 1, 2024

Video link

Minutes

- The vote:
  - Since we have a new proposal, the situation for the vote has been changed; let's discuss the new proposal first.
- PoC schedule sync-up:
- Interrupt suppressing inconsistency between multiple hit entries:
  - Agree the change
- New SRCMD format:
  - Andes and NVidia agree
  - Perrine will discuss this with the SiFive team and bring back the feedback before the Summit CN.
- The field of "version":
  - Bring the proposal on next meeting.

PoC Schedule Sync-up

- QEMU with IOPMP: postponed to 2024**Q4**
- M-mode library, lib-iopmp: still target to 2024**Q3** (if we can stabilize the spec soon)
- C/C++ model and the corresponding stimulus: by 2024**Q4**
- To provide SBI?
  - Andes, NVidia, and SiFive don’t provide SBI.
  - No plan to do it.

Interrupt suppressing Inconsistency

- Interrupt suppressing Inconsistency between multiple hit non-priority entries, example:
  - A write transaction hits two non-priority entries, entry(0) and entry(1), and
  - both entries don’t grant write permission; however
  - entry(0) suppresses the interrupt for write violation but entry(1) doesn’t, that is, entry(0).swie=1 and entry(1).swie=0.
- Will the IOPMP trigger the interrupt for the transaction?
  - According to the current spec, all hit entry(*i*), their swie’s are ORed together to obtain a final decision to suppress the interrupt. Thus, this example will not trigger any interrupt.
  - However, to trigger an interrupt is supposed to be more conservative from a security perspective.

Interrupt suppressing Inconsistency (cont.)

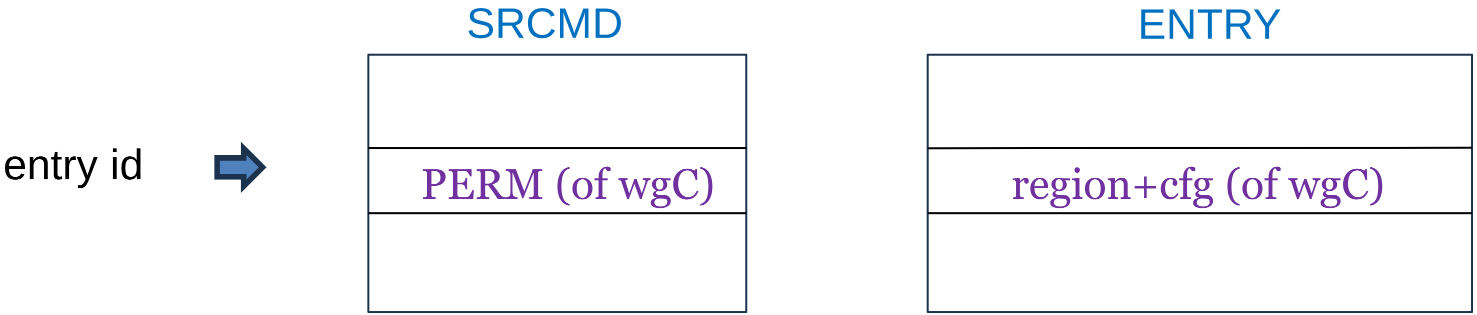
- The possible change if we update the spec according to
  - ERR_CFG.ie && !(entry(0).siwe || entry(1).siwe) // old ➔
  - ERR_CFG.ie && (!entry(0).siwe || !entry(1).siwe) // updated.
- That is, we OR all “to-trigger-intr” instead of “to-suppress-intr”

Alternative SRCMD Format

Example for proposed wgC-style program modeling

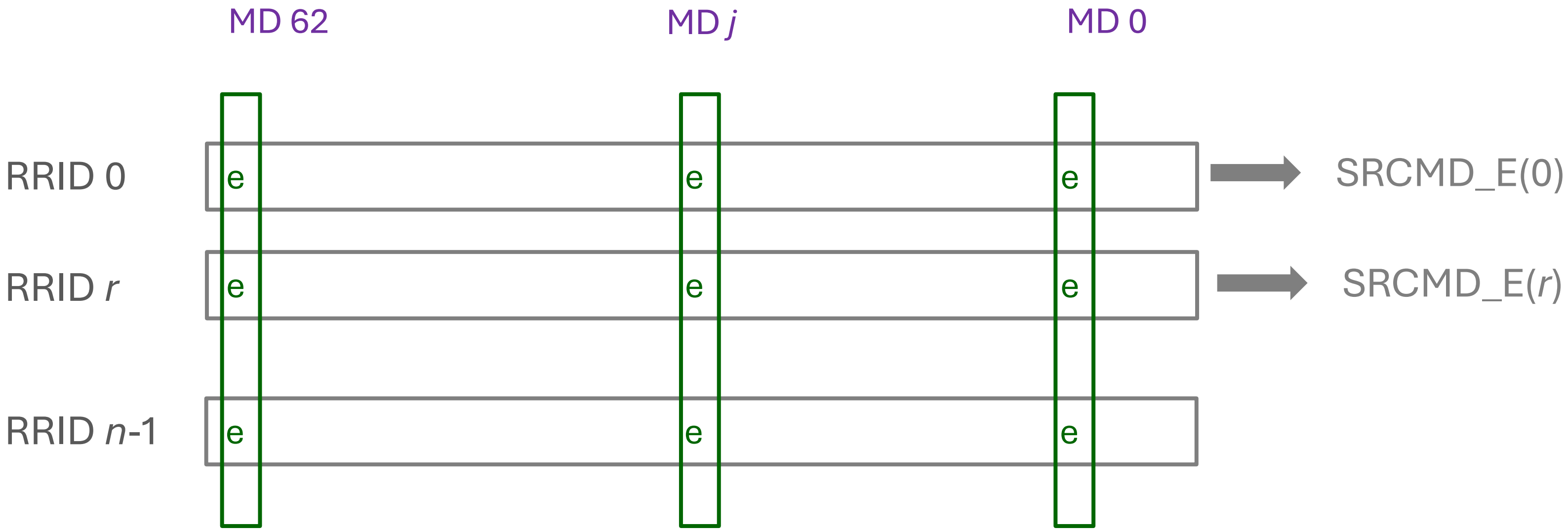
```
set_entry(entry entry_id, perm p, region r, cfg c) {
        SRCMD[entry_id] = p;
        ENTRY[entry_id].region = r;
        ENTRY[entry_id].cfg = c;
}
```

"perm" and the other configurations are set in two parallel arrays: ENTRY and SRCMD



Note: SRCMD_FMT=2; MDCFG_FMT=1; K=0 (in HWCFG0)

The Current SRCMD layout (indexed by RRID)



MD 62

MD *j*

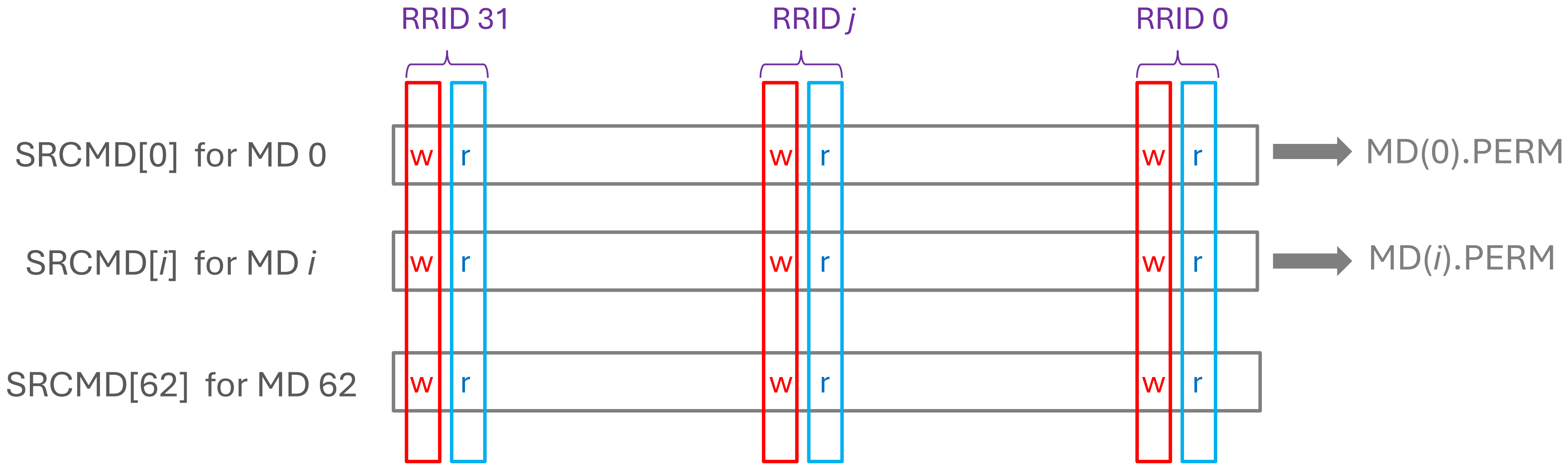
MD 0

RRID 0 e e e → SRCMD_E(0)

RRID *r* e e e → SRCMD_E(*r*)

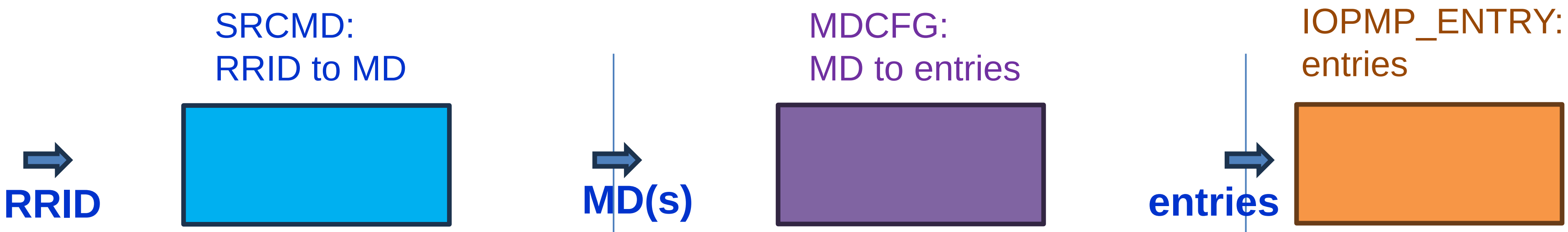
RRID *n*-1 e e e

The Alternative SRCMD layout (indexed by MD)



- ENTRY_CFG.r/w/a are ignored in this SRCMD layout

Proposed Tables



RRID

SRCMD:
RRID to MD

MD(s)

MDCFG:
MD to entries

entries

IOPMP_ENTRY:
entries

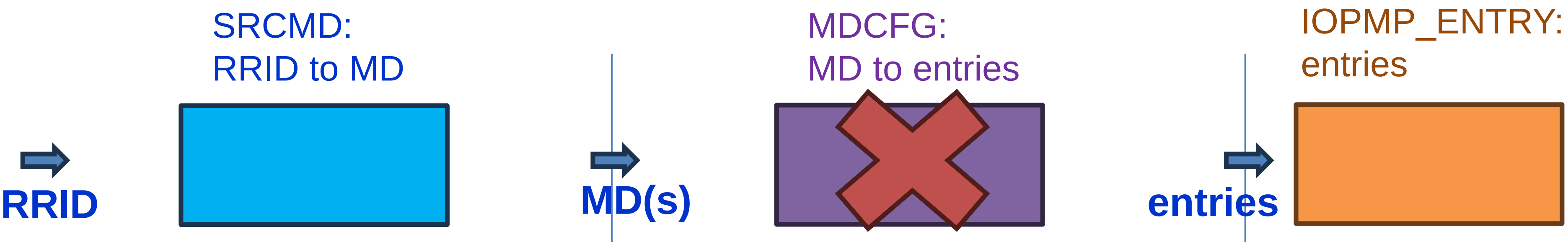
The association with RRID and MD(s),
lookup MD by RRID,
programmed by RRID or by MD:

- indexed by RRID
- indexed by MD
- fixed 1-1 mapping

Mapping an MD to entry(s);
The table is indexed by MD:

- full configurable
- 1 MD has *k* entries

The most wgC-style configuration



SRCMD:
RRID to MD

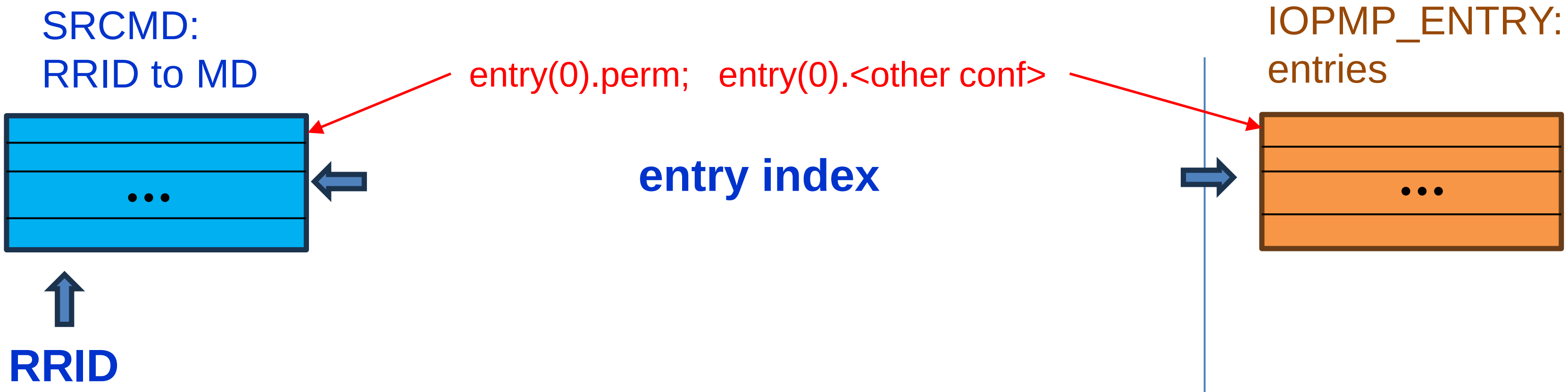
MDCFG:
MD to entries

IOPMP_ENTRY:
entries

Association with RRID and MD(s):
The table is indexed by MD
e.g., SRCMD_FMT = 2

1 MD equals 1 entry,
that is NO MD at all
e.g., MDCFG_FMT=1

That is,



SRCMD:
RRID to MD

entry(0).perm; entry(0).<other conf>

IOPMP_ENTRY:
entries

entry index

RRID

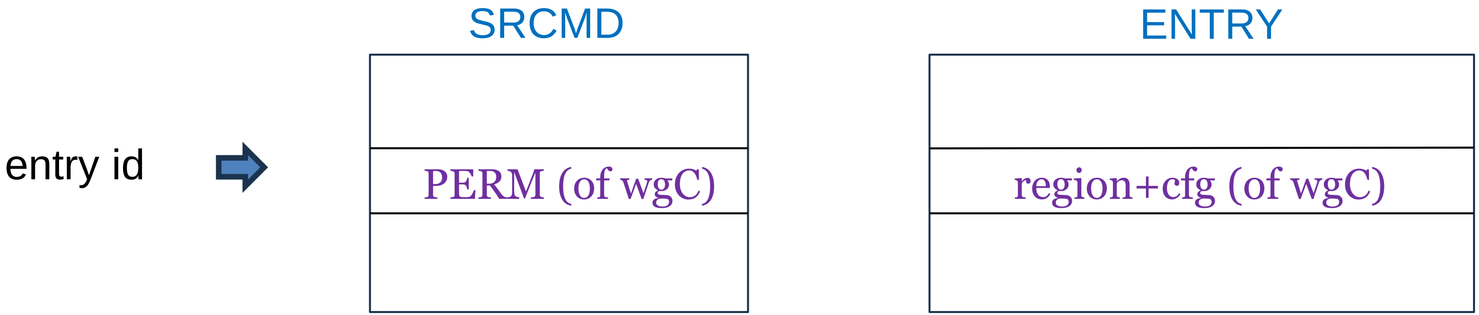
Association with RRID and entry(s):
The table is actually indexed by entry index
e.g., SRCMD_FMT = 2 and MDCFG_FMT=1

two parallel arrays

Example for proposed wgC-style program modeling

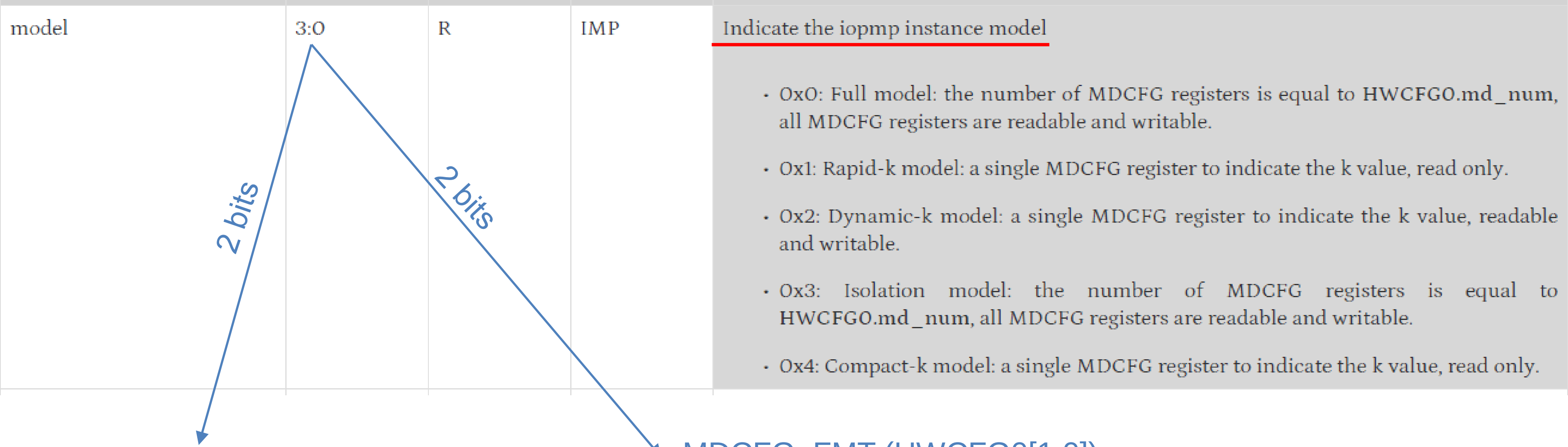
```
set_entry(entry entry_id, perm p, region r, cfg c) {
        SRCMD[entry_id] = p;
        ENTRY[entry_id].region = r;
        ENTRY[entry_id].cfg = c;
}
```

"perm" and the other configurations are set in two parallel arrays: ENTRY and SRCMD



Note: SRCMD_FMT=2; MDCFG_FMT=1; K=0 (in HWCFG0)

Register Adjust, HWCFG0.model


2 bits
2 bits

model	3:0	R	IMP	Indicate the iopmp instance model • 0x0: Full model: the number of MDCFG registers is equal to HWCFG0.md_num, all MDCFG registers are readable and writable. • 0x1: Rapid-k model: a single MDCFG register to indicate the k value, read only. • 0x2: Dynamic-k model: a single MDCFG register to indicate the k value, readable and writable. • 0x3: Isolation model: the number of MDCFG registers is equal to HWCFG0.md_num, all MDCFG registers are readable and writable. • 0x4: Compact-k model: a single MDCFG register to indicate the k value, read only.

SRCMD_FMT (HWCFG0[3:2]):

- 0: indexed by 1-1(isolation/compact)
- 1: indexed by RRID (full/rapid/dynamic)
- 2: indexed by MD (wgC)
- 3: reserved

MDCFG_FMT (HWCFG0[1:0]):

- 0: programmable by MDCFG (full/isolation)
- 1: same size MD, fixed (rapid-*k*/compact-*k*), no MDCFG
- 2: same size MD, programmable (dynamic-*k*), no MDCFG
- 3: reserved

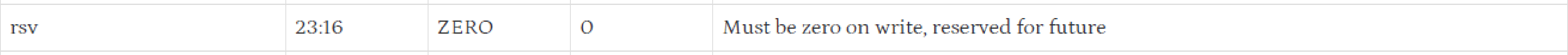
HWCFG0 will be adjusted to

- Adjust the 4-bit field model (HWCFG[3:0])
  - to 2-bit field SRCMD_FMT (HWCFG[3:2], RO):
    - ✓ 0: SRCMD is indexed by RRID (for full/rapid-*k* model)
    - ✓ 1: fixed 1-1 mapping (for isolation/compact-*k* model)
    - ✓ 2: SRCMD is indexed by MD (for wgC)
  - to 2-bit field MDCFG_FMT (HWCFG[1:0], RO):
    - ✓ 0: MD is configured by MDCFG (full/isolation); ignore field *K*
    - ✓ 1: same size MD, fixed (rapid-*k*/compact-*k*)
    - ✓ 2: same size MD, programmable (dynamic-*k*)
- Adjust the 8-bit field rsv (HWCFG0[23:16]), used on MDCFG_FMT=1 or 2
  - to 8-bit field *K* (WARL for MDCFG_FMT=2 and only when enable=0, otherwise RO):
    - ✓ the *k* value of rapid-*k*/compact-*k*/dynamic-*k* = *K* + 1.
    - ✓ For wgC-style model, *K*=0, that is, *k* value = 1.

SRCMD_FMT vs MDCFG_FMT

	SRCMD_FMT=0 (indexed by RRID)	SRCMD_FMT=1 (1-1 mapping)	SRCMD_FMT=2 (indexed by MD)
MDCFG_FMT=0 (by MDCFG)	full	isolation	no such model for now
MDCFG_FMT=1 (by fixed *K*)	rapid-*K*	compact-*K*	wgC (*K*=0)
MDCFG_FMT=2 (by programmable *K*)	dynamic-*K*	no such model for now	no such model for now

Adjust HWCFG0.rsv



rsv	23:16	ZERO	0	Must be zero on write, reserved for future

K	23:16	RW	IMP

MDCFG_FMT:

- 0: ZERO [0]; MD is configured by MDCFG
- 1: RO [*k* value, IMP]; No MDCFG
- 2: RW on enable=0; otherwise RO [*k* value, IMP] for dynamic-*k*; No MDCFG
- 3: ZERO (0); No MDCFG

Thank You

RISC-V®