IOPMP Task Group Meeting

August 3, 2023

Video link

Agenda

- Lite configuration extension:
- Parallel rule match extension:
- Error reporting on multi-fault
- Domain ID: naming issue

Two LC Proposals

- LC proposals: for smaller systems (especially for 32-bit control port)
  - Proposal-1: move the offset and occupy smaller space
  - Proposal-2: not move the offset
- Notation Issue:
  - XLEN may not make sense for IOPMP
  - Control Port Data Width: CPDW (by one transaction)
  - Input Port Address Width: IPAW
- If #(MDs) is reduced and the lock bit is moved, we need a bit to tell programmers.

32 Bits and 64 Bits Support

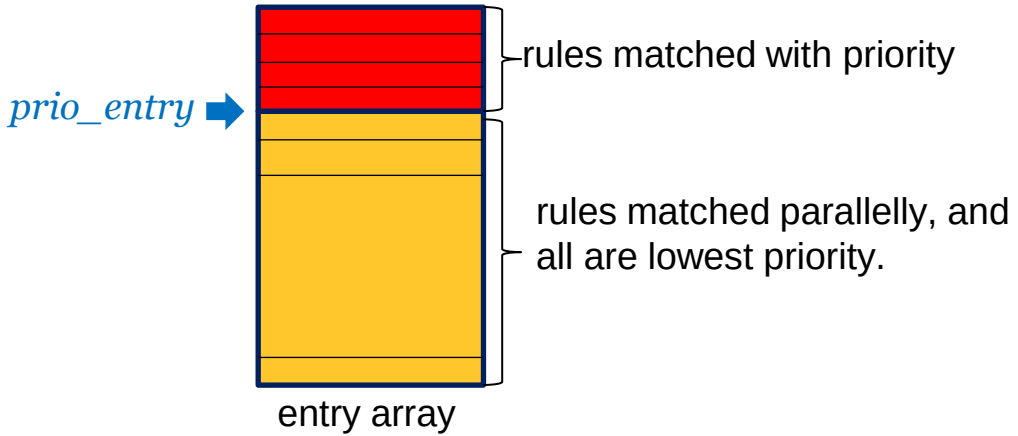
			IPAW	
			32	64
CPDW	32	31 MDs	LC	V2 (?)
		63 MDs	V1 (?)	V3 needed
	64	63 MDs	NA	Ready

64-bit Atomic Write on 32-bit CPDW

- A 32-bit scratch register
- Every 64-bit register:
  - separated to two 32-bit registers: odd and even
  - placed nature align (8-byte align)
- To write a 64-bit register:
  1) Write the value for its even register to the scratch register
  2) Then, write to its odd register. Its even register is loaded from the scratch register synchronously.
- 32-bit register will NOT be allocated in the odd register.

To Support Parallel Rule Match

- *prio_entry*, indicates how many entries match according their priorities
- *prog_prient*: indicate if *prio_entry* is programmable.
  - If the implementation allows *prio_entry* programmable, *prog_prient* =1 on reset.
  - When clean *prog_prient*, it is stickily to zero until reset.
  - If *prio_entry* is read-only, *prog_prient* is wired to 0.



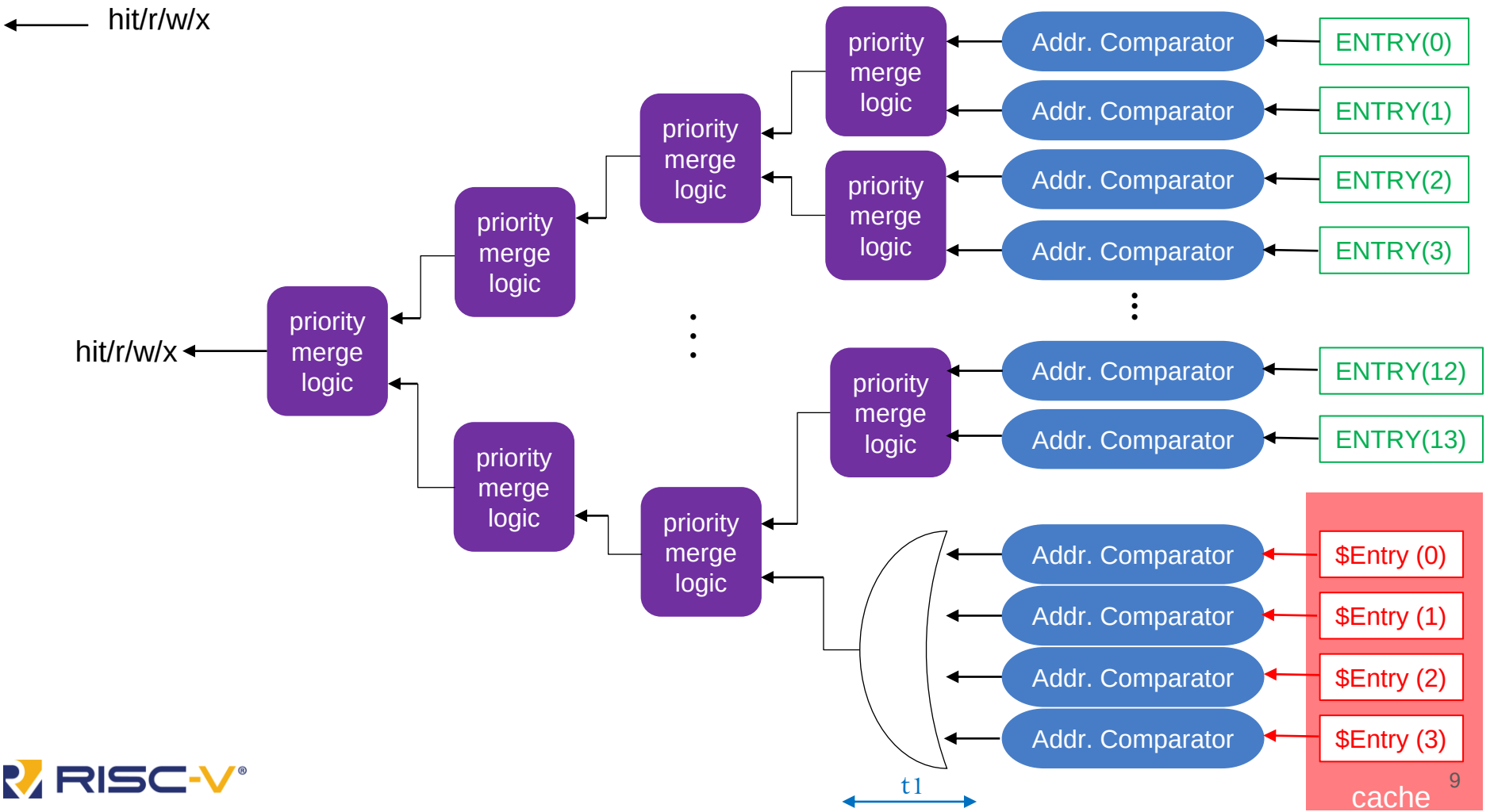
The Number of Priority Rules

HWCFG1			
Field	Bits	R/W	Description
tor_en	0:0	R	Indicate if TOR is supported
sps_en	1:1	R	Indicate the secondary permission settings is supported
user_cfg_en	2:2	R	Indicate the if user customized attributes is supported
prog_prient	3:3	W0	A sticiky bit to indicate if prio_entry is programmable. Reset to 1 if the implementation supports programmable prio_entry, otherwise, wired to 0.
model	7:4	R	Indicate the iopmp instance model
prio_entry	31:16	WARL	Indicate the number of entries matched with priority. These rules should be placed in the lowest order. Within these rules, the lower order has a higher priority.

Wired to: 0➔WG-mode; entry_num➔all-priority-mode; non-zero➔typical case
WARL: accept some legal value(s). (RO is a kind of WARL)

A Reference Implementation

- 128 Rules:
  - 14 priority entries for the most sensitive data
  - 114 The others are equal priority (parallel rule match)
- Single cycle in the typically cases (cache hit)
- 4 Cached entries
- Concern: performance and area


hit/r/w/x
hit/r/w/x
priority merge logic
priority merge logic
priority merge logic
priority merge logic
priority merge logic
priority merge logic
priority merge logic
priority merge logic
priority merge logic
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
Addr. Comparator
ENTRY(0)
ENTRY(1)
ENTRY(2)
ENTRY(3)
ENTRY(12)
ENTRY(13)
$Entry (0)
$Entry (1)
$Entry (2)
$Entry (3)
t1
cache

Domain ID (only Cover Source Sides' ID)

- The IDs to be unified to Domain ID are source-sided:
  - MTT, SDID, WID, SID, …
- Memory Domain is destination-sided, and not included in the name:
  - Suggest: change the name in order to distinguish it from the destination-side IDs.

Thank You