IOPMP Task Group Meeting

April 11, 2024

Video link

Minutes

- Released draft 6, change log:
- Clarification: MDSTALL.is_stalled
- Issue: what if the rule of MDCFG.t is breached?

Draft 6, Change log

- Change log:
  - Rename SID to RRID
  - Per-entry interrupt control
  - Per-entry bus error control
  - Multi-fault record
  - Refine, clarify, and typo-fixup

Clarification: MDSTALL.is_stalled

- What if writing a non-zero value to MDSTALL, but no RRID is selected?
  - **MDSTALL.is_stalled** should be “1” , because
  - It can avoid infinite polling of this flag because it is never asserted.
  - That is, only writing a zero to MDSTALL(H) (resume command) can de-assert “**is_stalled**.”

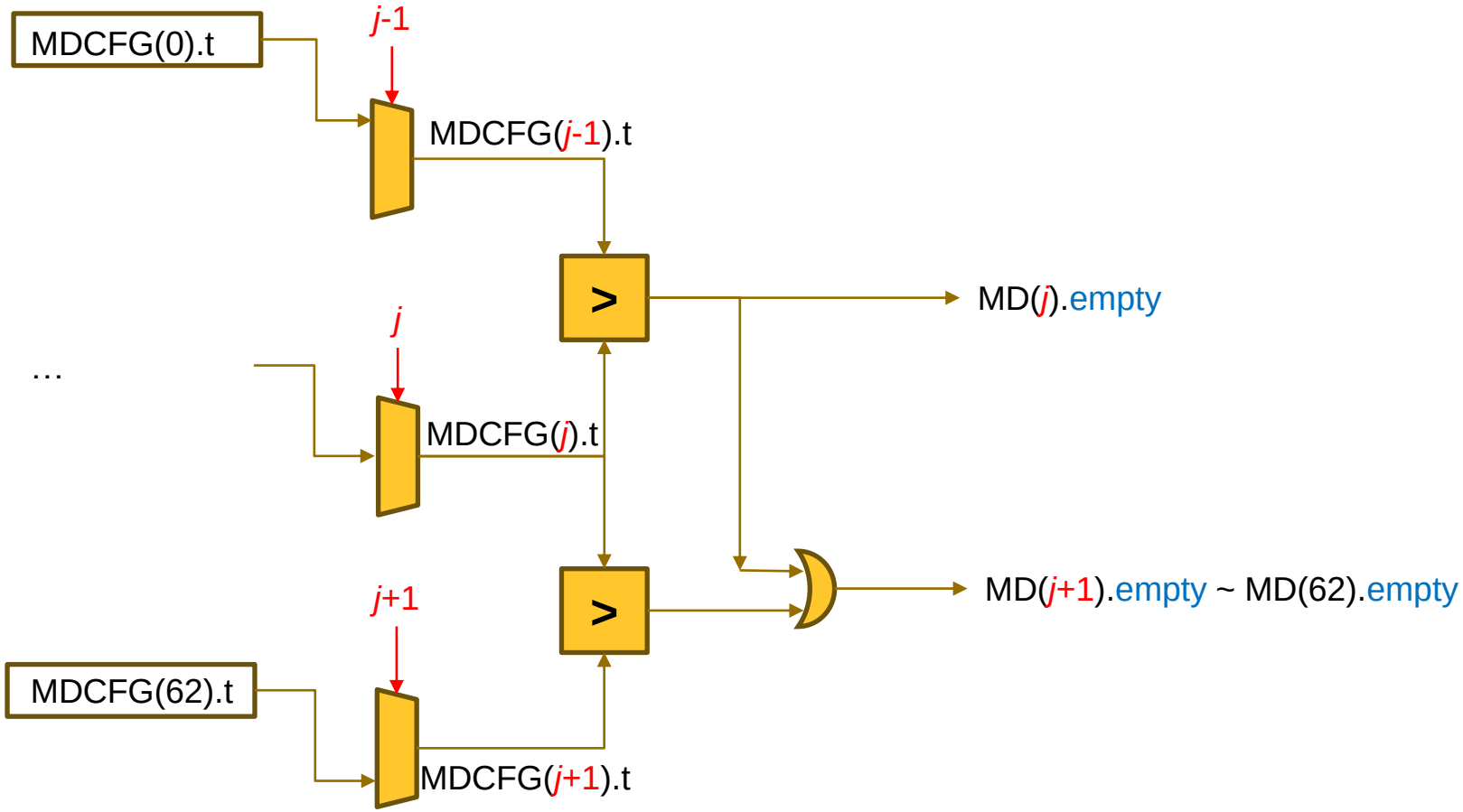
A reference flow to program an IOPMP

```
Step 1.1: write MDSTALL once // exactly once
Step 1.2: write RRIDSCP zero or more times
Step 1.3: poll until MDSTALL.is_stalled == 1 // ensure all stalls
take effect
Step 2: update IOPMP's configuration
Step 3.1: write MDSTALL=0 // resume all transactions
Step 3.2: poll until MDSTALL.is_stalled == 0  // ensure all resumes
take effect
```

Some steps may be skipped according to implementations or use cases.

Issue: what if MDCFG.t is NOT monotonically incremental?

- In full-model or isolation-model, what if **MDCFG**[*j*].**t** > **MDCFG**[*j*+1].**t** when writing **MDCFG**[*j*] or **MDCFG**[*j*+1]?
  - Policy:
    1) Implementation-dependent, no specific HW behavior is required
    2) List one or a few options to implement
  - Possible options:
    1) Ignore the write causing the case // some programming order is required.
    2) For $j<m\leq 62$, fix **MDCFG**[*m*].**t** = MIN { **MDCFG**[*k*].**t** | $0\leq k\leq m$ }
    3) Modify **MDCFG**[*j*+1].**t** = **MDCFG**[*j*].**t**. // Channing will post the slides separately
    4) Mark MD(*j*+1) to MD(62) empty (have no entry) // possible imp on next page



On writing MDCFG[*j*]