TDA4VE Inline ECC Cache Dirty Bits Issue

Root Cause & Workaround

TI Confidential – NDA Restrictions

Inline ECC Cache Issue - Root Cause & Workaround

Issue Summary

False ECC errors reported by DDR inline ECC logic despite no actual data corruption. Depending on timing, this manifests as 2-bit errors (read data zeroed out), 1-bit errors (data incorrectly corrected), or silent corruption (no error flagged). If the read is part of a read-modify-write, corrupted data is written back to DDR.

Root Cause

A one-cycle race condition in the inline ECC cache dirty-bit tracking:

1. Cache line holds ECC for Block A (dirty + complete)
2. New write for Block B evicts Block A - complete flag resets immediately, dirty bits reset one cycle later
3. Second write for Block B arrives in that one-cycle gap
4. Stale dirty bits from Block A make it appear the ECC is fully written
5. ECC marked complete prematurely - later reads trust invalid ECC

All conditions must align: cache line dirty, new block evicts old, second write to same new block next cycle, stale dirty bits cover unwritten ECC portions.

Workaround (Immediate)

Disable write allocation: Set `wr_alloc = 0` in ECC Control Register
AND
Do not allocate cache lines via RouteID mapping (ECC Cache RouteID Index/Value registers)
Note: May have performance impact depending on traffic pattern

Status

Use SW workaround for affected devices.

TI Confidential – NDA Restrictions