SYNOPSYS®
Silicon to Software™

DWC_mshc

STAR #9001226488 - Early Transfer complete Interrupt issue

Mobile storage Team

08/17/2017

CONFIDENTIAL INFORMATION

The following material is confidential information of Synopsys and is being disclosed to you pursuant to a non-disclosure agreement between you or your employer and Synopsys. The material being disclosed may only be used as permitted under such non-disclosure agreement.

IMPORTANT NOTICE

In the event information in this presentation reflects Synopsys’ future plans, such plans are as of the date of this presentation and are subject to change. Synopsys is not obligated to develop the software with the features and functionality discussed in these materials. In any event, Synopsys’ products may be offered and purchased only pursuant to an authorized quote and purchase order or a mutually agreed upon written contract.

Agenda

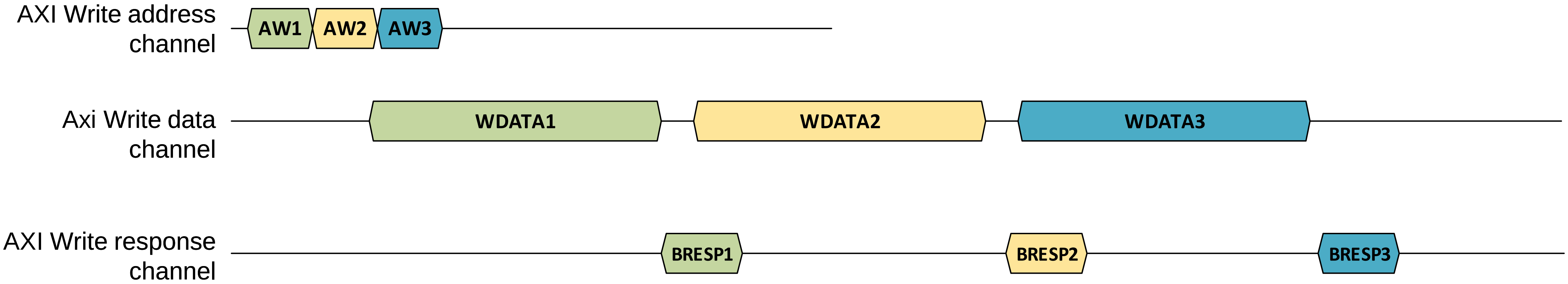
- STAR #9001226488
  - Scenario description
- Impacted Configurations
- Workaround and RTL fix

STAR #9001226488
Early transfer complete (XFER_COMPLETE) interrupt issue

- AXI Slave sends write response after 64+ AXI clock cycles from last dword of same transfer

Transactions Shown here are on AXI interface

OKAY Scenario



AXI Write address channel: AW1, AW2, AW3

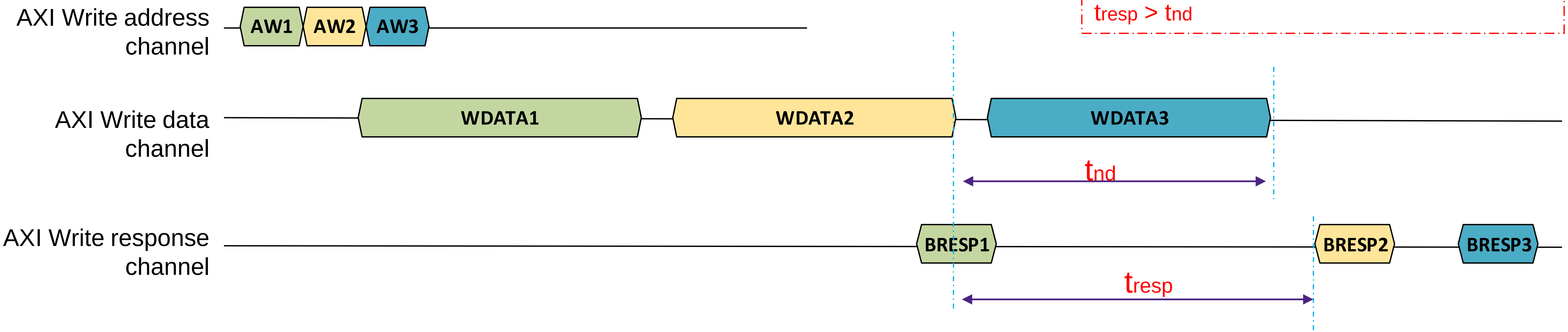
Axi Write data channel: WDATA1, WDATA2, WDATA3

AXI Write response channel: BRESP1, BRESP2, BRESP3

Issue Scenario

Problem occurs when the response for a write data transfer is delayed such that it received after next data transfer

$t_{resp} > t_{nd}$



AXI Write address channel: AW1, AW2, AW3

AXI Write data channel: WDATA1, WDATA2, WDATA3

t_{nd}

AXI Write response channel: BRESP1, BRESP2, BRESP3

t_{resp}

Scenario description

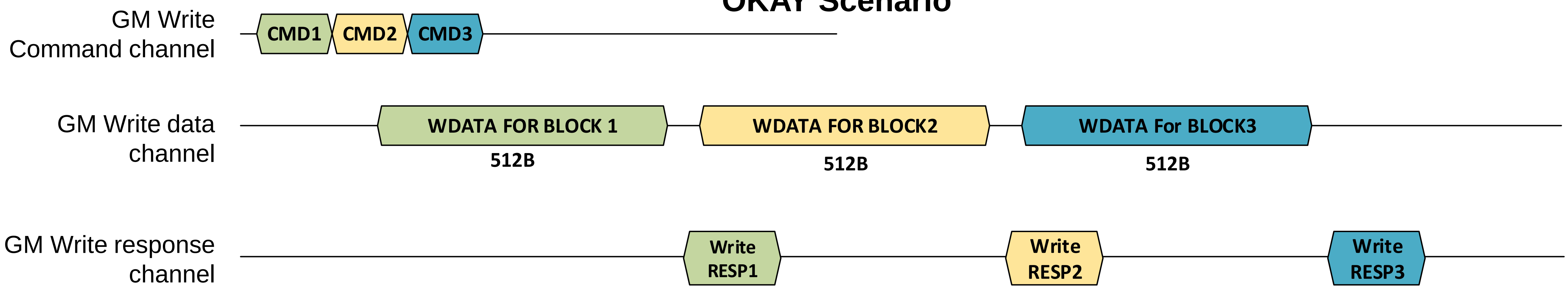
- Controller Block size is small = 6 Bytes
- eMMC Read operation

Impact of this issue

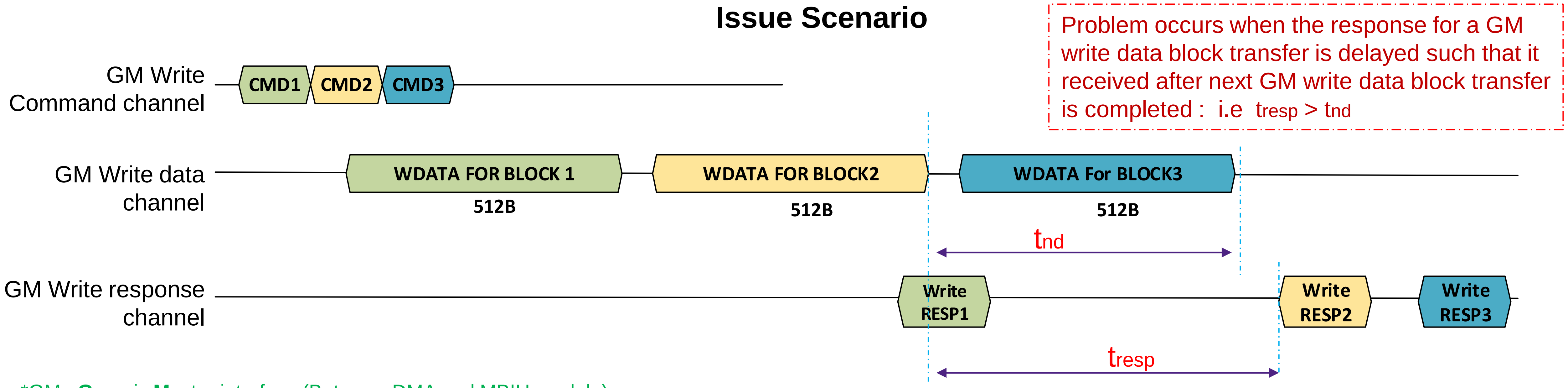
- Due to this issue : Additional decrement block messages are pushed by DMA
- This causes Transfer complete interrupt (XFER_COMPLETE) to be generated early
  - DMA has not yet completed the transfer
  - But register block count (regb_blk_cnt) goes to zero early
- Since transfer complete is generated (early)
  - SW can issue next transfer
  - This next transfer can corrupt ongoing DMA transfer
  - However this issue **does not cause data corruption** if a new transfer is not started immediately.
- Impacts ADMA2
  - Does not impact SDMA
  - Does not impact CMDQ
  - Partially impacts ADMA3 : Impacts only the last ADMA3 transfer (ID descriptor end)
- Does not impact Write outstanding=4 configuration (DWC_MSHC_AXI_MAX_WR_REQUESTS==4)
- Next slide shows how this can happen when block size is 512Bytes

Transactions Shown here are on GM* interface

OKAY Scenario



Issue Scenario



Problem occurs when the response for a GM write data block transfer is delayed such that it received after next GM write data block transfer is completed : i.e $t_{resp} > t_{nd}$

*GM : Generic Master interface (Between DMA and MBIU module)

Note:

- A 512B request on GM interface is spit into 4 AXI Write burst transfer (AWLEN=16)
- A GM Write block data transfer is considered completed only after data for all the 4 (split) AXI transfers are sent to slave.
- And a GM Write response is generated only after response for all the 4 AXI Writes are received.

Impacted Configurations

Impacted Configurations

- AXI Interface Configurations with outstanding write requests more than 4:
  (DWC_MSHC_MST_IF_PRESENT==1) and
  (DWC_MSHC_MST_INTERFACE_TYPE==0) and
  (DWC_MSHC_AXI_MAX_WR_REQUESTS > 4)

- OR

- (DWC_MSHC_MST_IF_PRESENT==1) and
  (DWC_MSHC_MST_INTERFACE_TYPE==0) and
  Block size used is less than or equal to 256Bytes

Work Around

- Software
- Hardware or RTL Fix

Workaround : Software

MBIU_CTRL_R (Offset : P_VENDOR_SPECIFIC_AREA[11:0] + 0x10)

- Bits [26:24] of this register can be used to control AXI's Write outstanding limit.
- This is an internal debug register and its description is not present in databook
- These bits should be programmed during initialization
- These bits should not be changed dynamically and on-the-fly

BLOCK_SIZE = 512Bytes : Set MBIU_CTRL_R[26:24] to 0x3 (Write Outstanding 4)

BLOCK_SIZE = 256Bytes : Set MBIU_CTRL_R[26:24] to 0x1 (Write Outstanding 2)

BLOCK_SIZE <=128Bytes : Set MBIU_CTRL_R[26:24] to 0x0 (Write Outstanding 1)

- This only modifies AXI Write outstanding. However AXI read outstanding is unchanged

Assessment for Workaround Option1

- Option1 can be hard for the software to implement as it needs to guess the wait time.
- If Max wait time is used then there will be a performance loss.
- At the moment we are not exploring this as we are focusing on Option2 and RTL Fix.

Workaround : Hardware

- Customer has to confirm if the scenario of $t_{resp} > t_{nd}$ is valid in their use case
  - Kindly consider the system latency
  - And the AXI slave write response behavior

- Customer has to confirm if their application will use small block sizes
  - Block sizes of less than 16bytes

- Synopsys has fixes for this STAR# 9001226488 and patch release **1.70a-ea02** is readily available

END