High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

Webinar

Objectives

Understand how the High Bandwidth Memory (HBM2), found in Intel® Stratix® 10 MX devices, offers a compelling solution for the most demanding memory bandwidth requirements

Parameterize and integrate the HBM2 IP into an Intel Quartus® Prime project design

Verify the functionality and check the performance and efficiency of the integrated HBM2

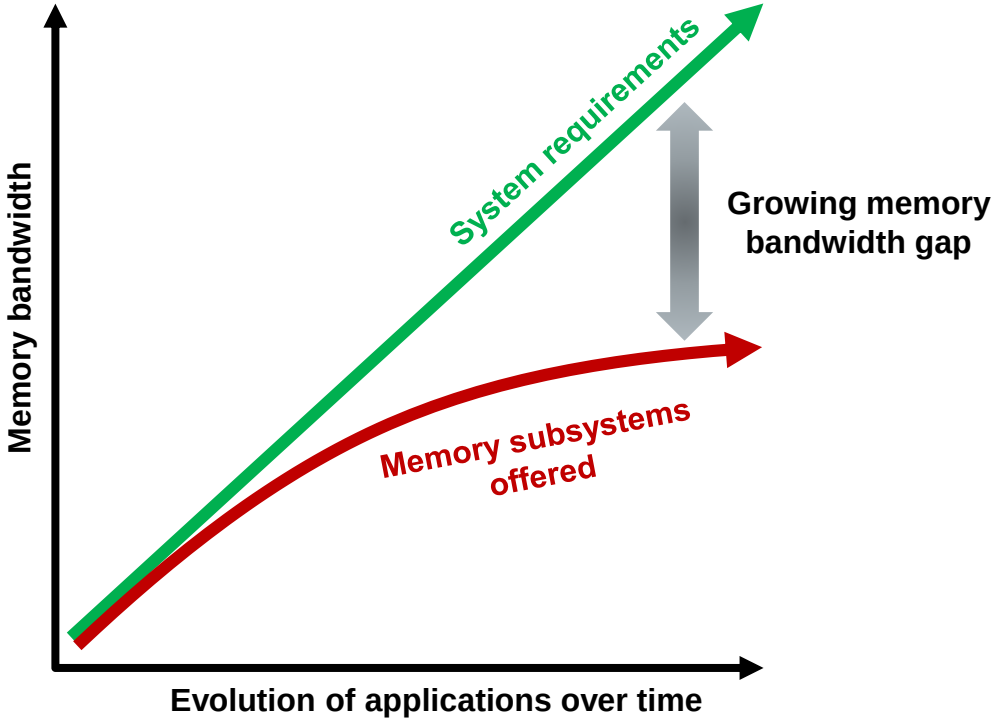
Agenda

- Introduction to HBM2
- HBM2 architecture
- HBM2 controller (HBMC) features
- User interface to HBM2
- HBM2 IP implementation
- Performance and efficiency

High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

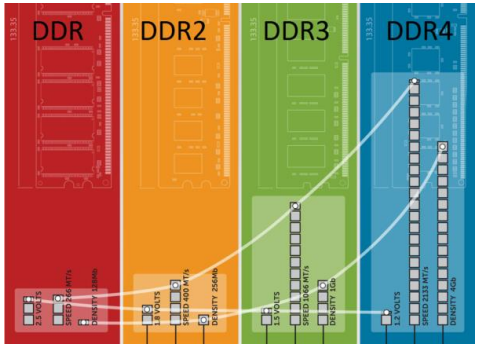
Introduction to HBM2

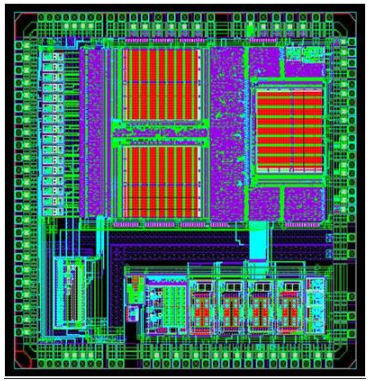
Need for Memory Bandwidth Is Critical



Key Challenges to Meeting Memory Bandwidth

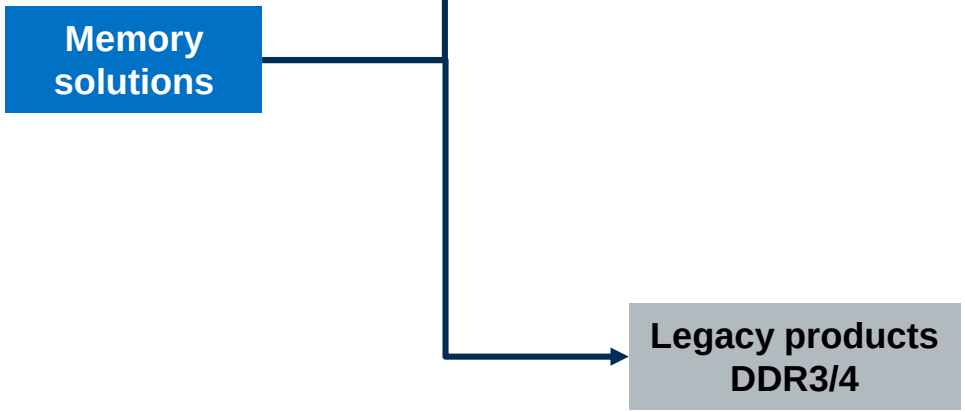
1. Uncertain DDR roadmap
2. Memory bandwidth is I/O limited
3. Flat system-level power budgets
4. Limits to monolithic memory integration





Innovation needed to meet high end memory bandwidth requirements

New DRAM Memory Landscape



Micron

Hybrid Memory Cube (HMC)

MoSys

Bandwidth Engine

Serial interface (external components)

Parallel interface (integrated into FPGA/ASIC)

SK hynix

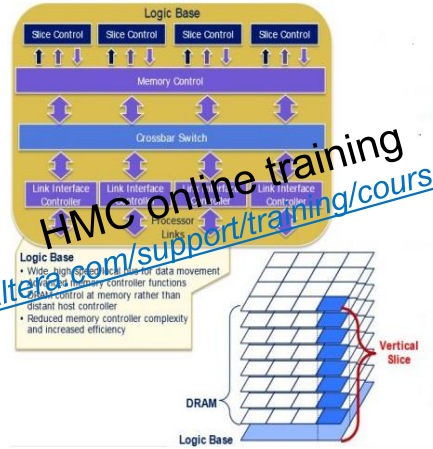
SAMSUNG

- High Bandwidth Memory (HBM)
- Wide I/O Memory (WIO)

Parallel standards supported by multiple vendors

Supported Next Generation DRAM: A Closer Look

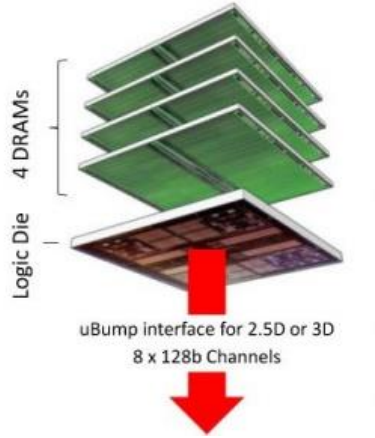
Hybrid Memory Cube (HMC) (HMCC)



HMC online training
https://www.altera.com/support/training/course/ohmc101.html

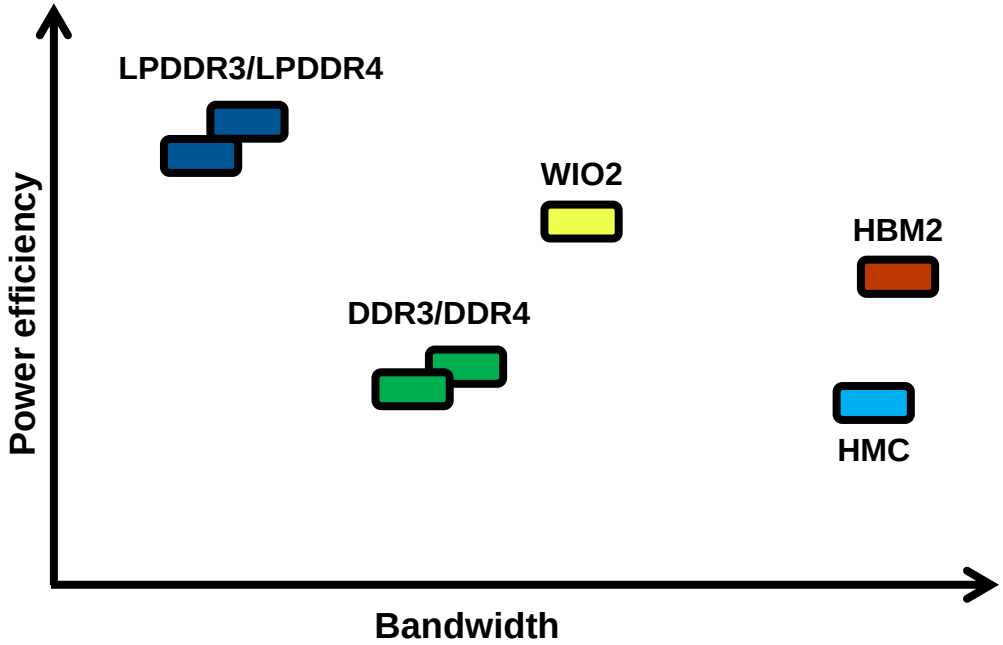
- 3D Through Silicon Via (TSV) DRAM Stack
- Up to **320 GB/s**
- Up to 30 Gbps per serial link
- Discrete memory cores
- Controller die at base
- 4 serial links, 64 XCVR lanes
- High power
- Supported in Intel® FPGA Arria® 10 and Stratix® 10 devices

High Bandwidth Memory Gen 2 (HBM2) (JEDEC)



- 3D TSV DRAM stack
- Up to **256 GB/s** per stack at up to **1 GHz**
- 1024-bit parallel interface
- Up to 2 Gbps per I/O
- Controller in host
- Low power
- Built into Intel FPGA Stratix 10 MX devices as a System in Package (SiP) memory

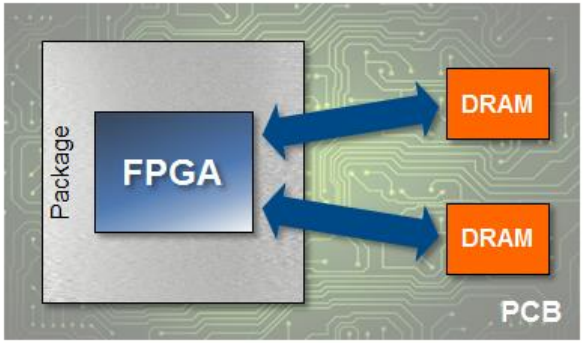
DRAM Solutions by Bandwidth and Power



HBM GEN2 offers optimum bandwidth/power

Discrete vs. SiP Memory

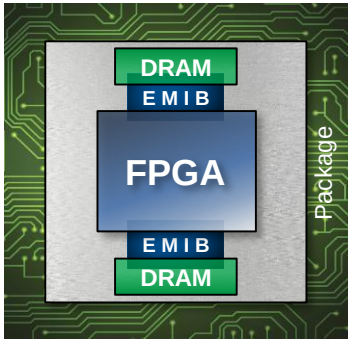
“Far” memory with discrete DRAM



- ✘ Lower bandwidth
- ✘ Higher power
- ✘ Larger footprint

Cannot meet requirements of next-generation applications

“Near” memory with DRAM SiP



- ✓ Highest bandwidth
- ✓ Lowest power
- ✓ Smallest footprint

Meets the memory bandwidth needs of next-generation applications

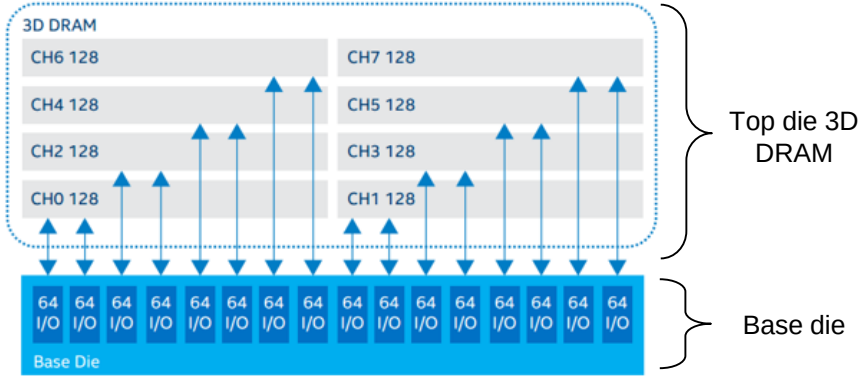
High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

HBM2 Architecture

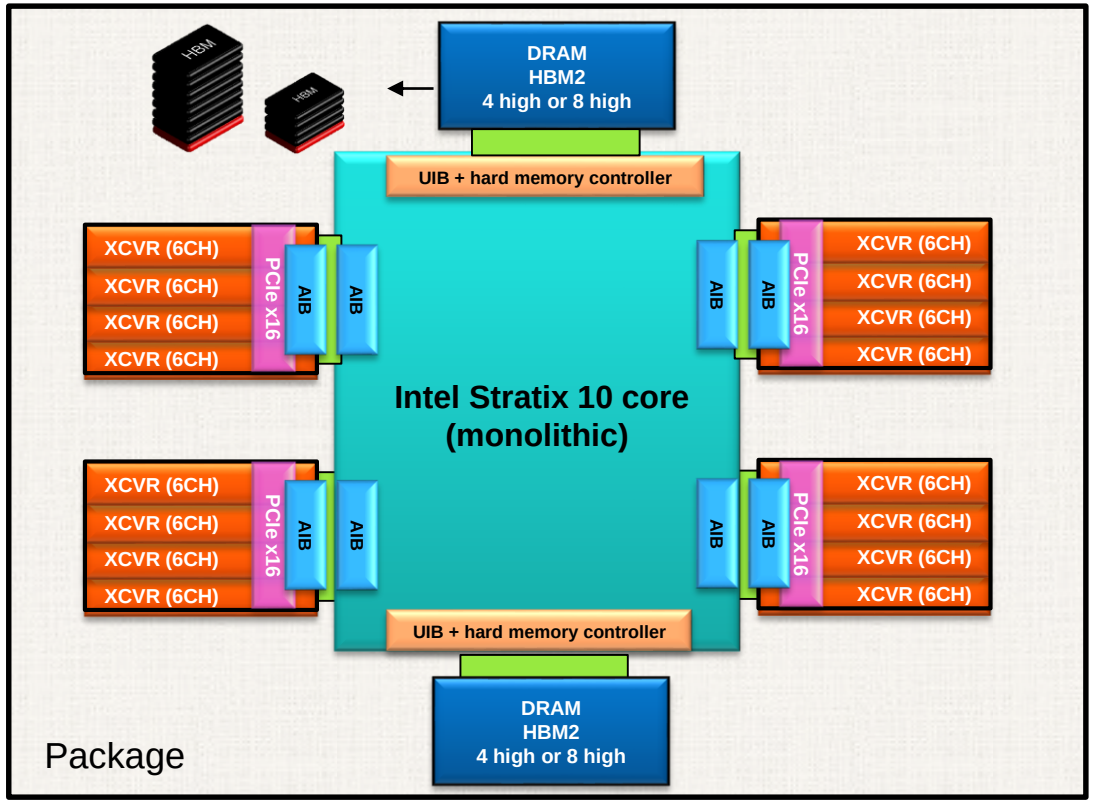
Integrated 3D HBM2 DRAM Stackup

- Top die: 3D DRAM using TSV
  - 4 GByte / 4 high or 8 GByte / 8 high
  - 4 to 8 Gbit capacity per channel
- Base die: I/O Interface
  - 8 independent physical I/O channels
  - 16 independent **pseudo**-channels (2 pseudo-channels per physical channel)
  - 64-bit wide I/O per pseudo-channel
- Ultra-high parallel bandwidth
  - 2048 Gbit/s (256 GByte/second)
    - 128 bit * 8 channels * 2 (for DDR) * 1GHz

HBM2 DRAM (4 Gbyte / 4 High)



Intel® Stratix® 10 MX Device Layout



AIB

Interface bus

Embedded Multi-die Interface Bridge (EMIB)

UIB

Universal Interface Bus + hard memory controller

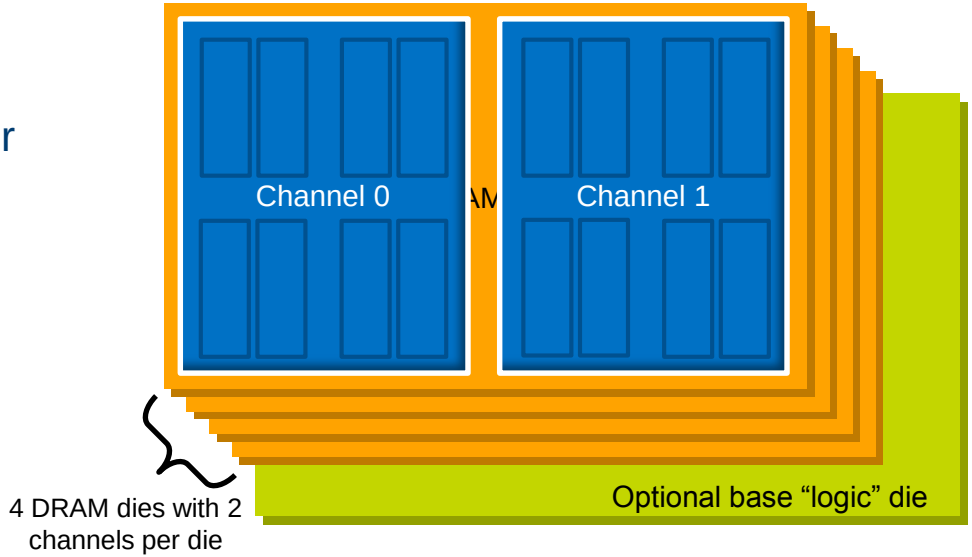
Illustration only, not drawn to scale

HBM2 System in Package (SiP) Components

- HBM2 memory stack
- Universal Interface Block Sub System (UIBSS)
- Embedded Multi-die Interconnect Bridge (EMIB)
  - Silicon bridge to connect FPGA I/Os to memory device
- User logic interface to access memory
- Arm* AMBA* 4 AXI interface soft logic adapter *(optional)*
- HBM2 controller (HBMC), part of the UIBSS

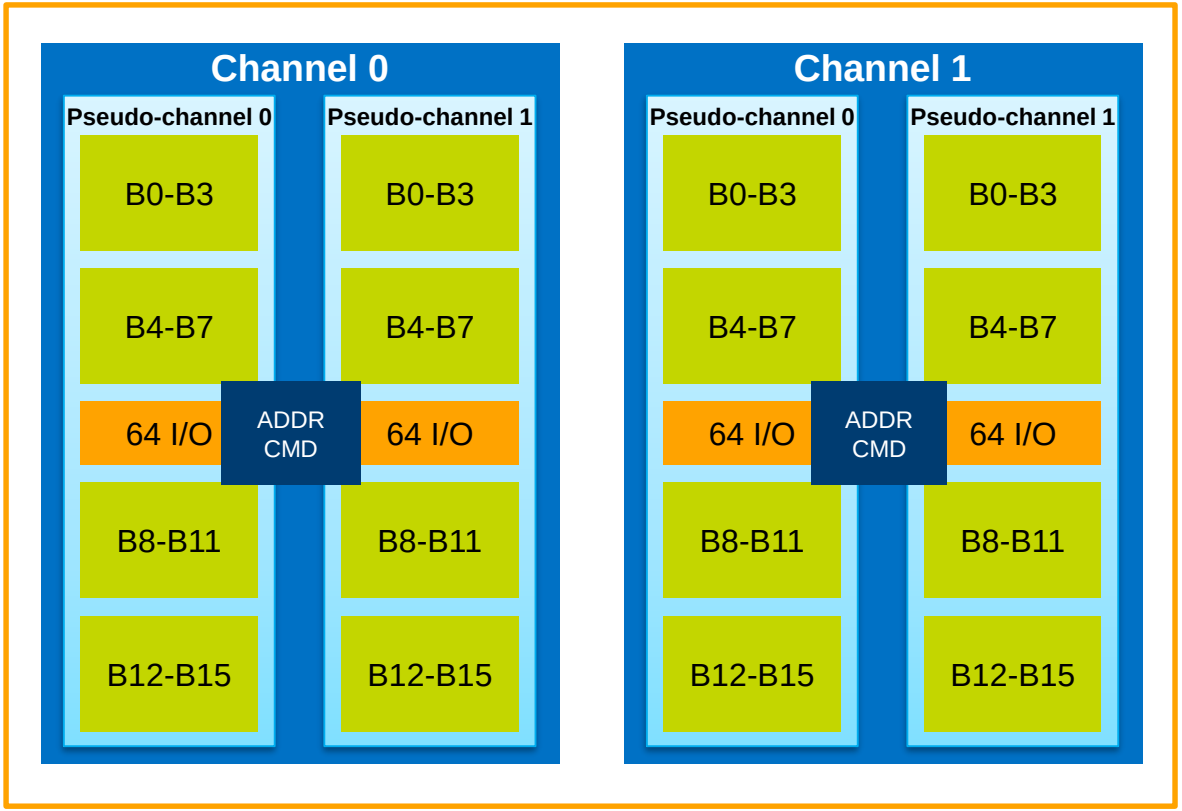
HBM2 Architecture Detail

- HBM2 DRAM organized in stacks
  - 4 DRAM dies per stack + base die or
  - 8 DRAM dies per stack + base die
- Each DRAM die has
  - Density of 8 Gb (1 GB)
  - 2 channels



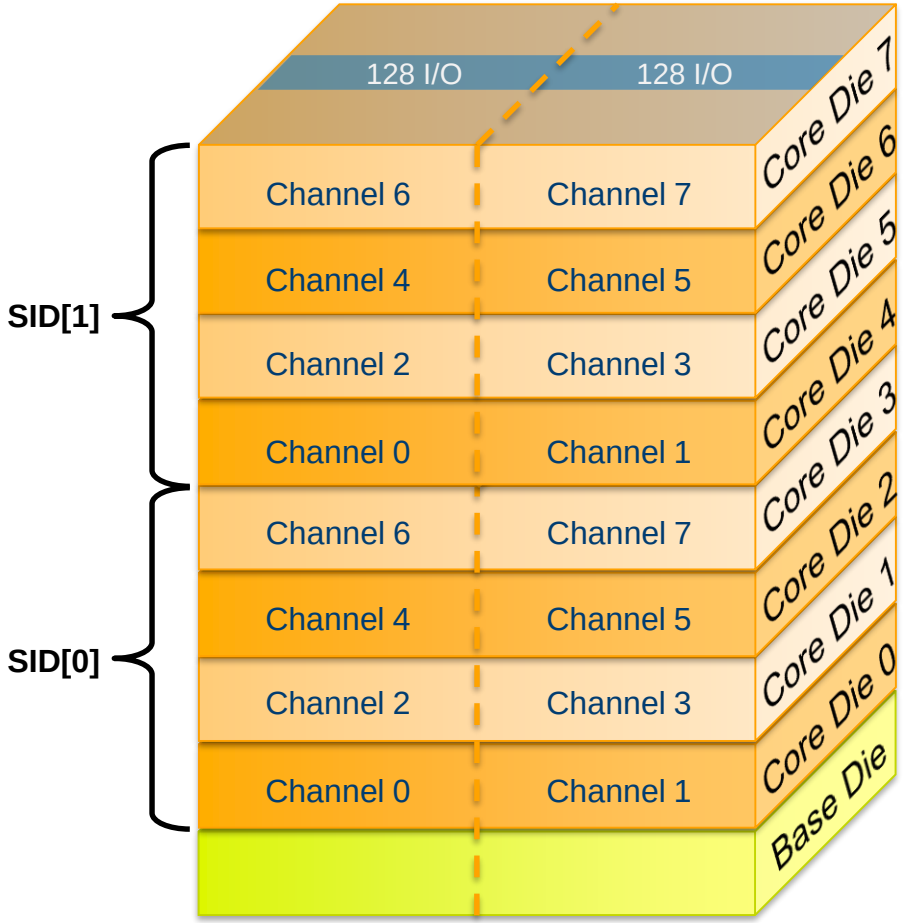
Channel Detail

- Each channel has:
  - 2 pseudo-channels
  - 128 data I/Os
    - 64 per pseudo-channel
  - Independent address/command bus
  - Access to an independent set of banks
  - 4 Gb density (4H) or 8 Gb density (8H)



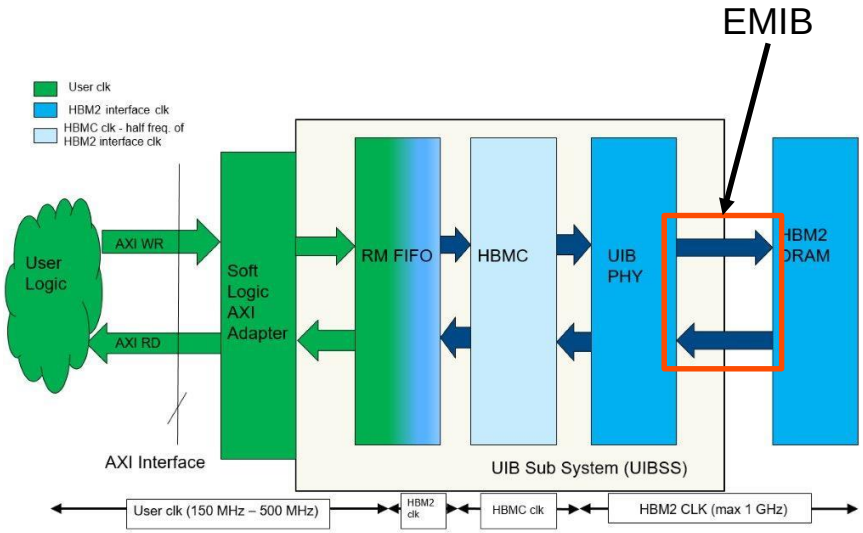
HBM2 Stack

- Increased stack height allows for greater density
  - Still limited to 8 HBM2 channels
- Stack ID (SID) distinguishes between upper/lower dies



UIBSS Components

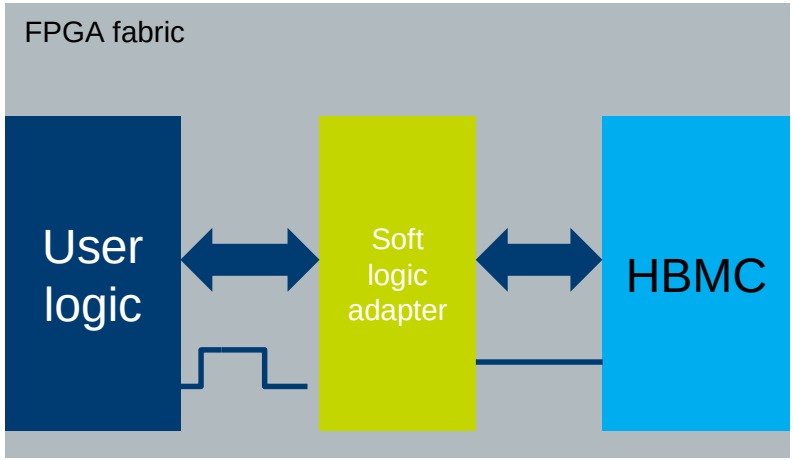
- UIB PHY
  - Contains logic and I/O buffers to drive the signals over EMIB to perform memory operations
- Rate matching (RM) FIFOs
  - For clock crossing from user clock domain to PHY clock domain
- HBM2 Memory Controller (HBMC)
  - Performs read/write transactions per memory protocol specification
  - Performs memory housekeeping operations to ensure proper data operation and retention



Arm* AMBA* 4 AXI Interface Soft Logic Adapter

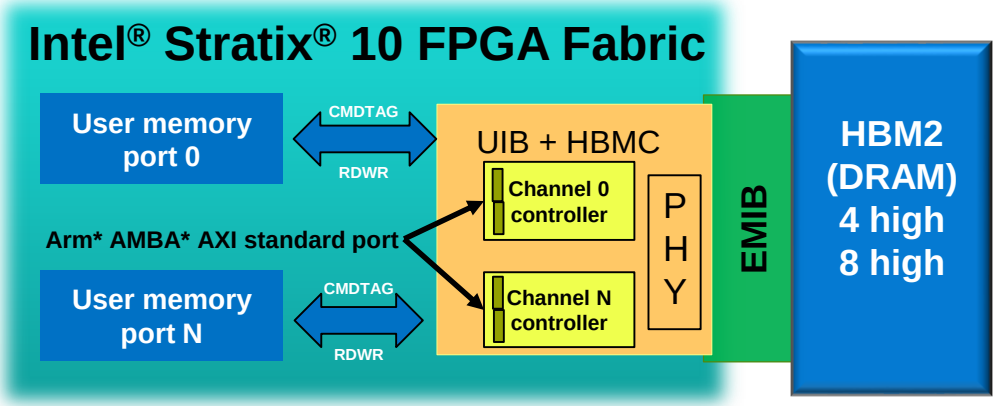
Optionally include this adapter to gate handshaking VALID signals from HBMC

- HBMC *(discussed next)* connects to user logic and HBM2 stack via Arm AMBA 4 AXI interface (256 bit read/write data channels)
- User logic may not always be ready to accept data from HBMC
- Signals gated by adapter: **AWVALID**, **WVALID**, **ARVALID**
- Also stores write response (**B**) and read data (**R**) channel when **READY** deasserted
- Not needed if user logic always ready to receive data



Dedicated Hard HBM2 Controller (HBMC)

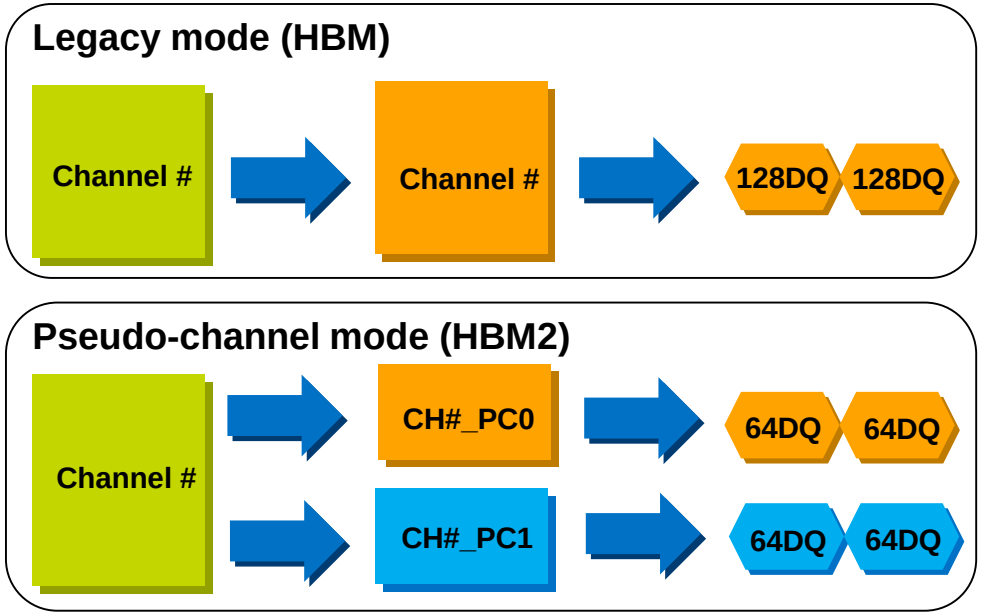
- Higher performance, higher efficiency, and lower power relative to soft controller
- Supports 4 high and 8 high HBM2 stack
- Independent controller per physical channel (always 8 channels)
  - 16 pseudo-channels have user interface via two Arm* AMBA* 4 AXI interface ports per channel
  - 32 and 64 byte access
  - One Arm AMBA APB interface sideband



Meets HBM2 JEDEC specification

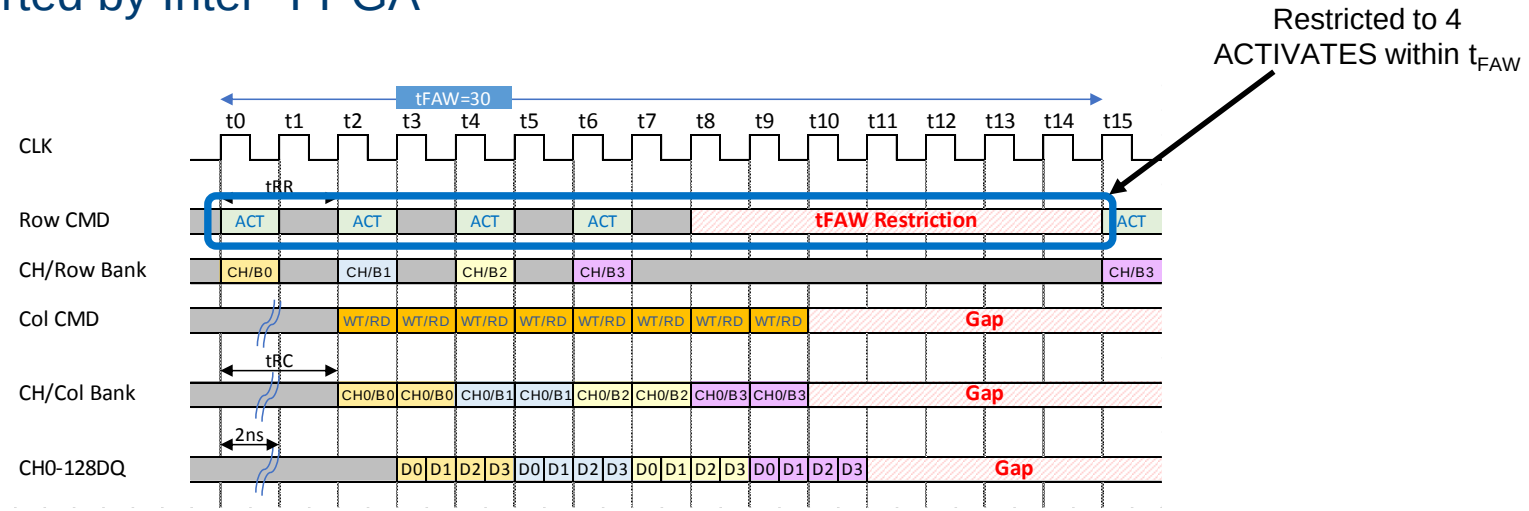
HBM2 Pseudo-Channel Mode

- Only available in HBM2
- Each channel consists of 2 pseudo-channels
- 64 DQ per pseudo-channel
  - Address/command bus is shared
  - Decode and execute commands individually
  - Helps maximize throughput



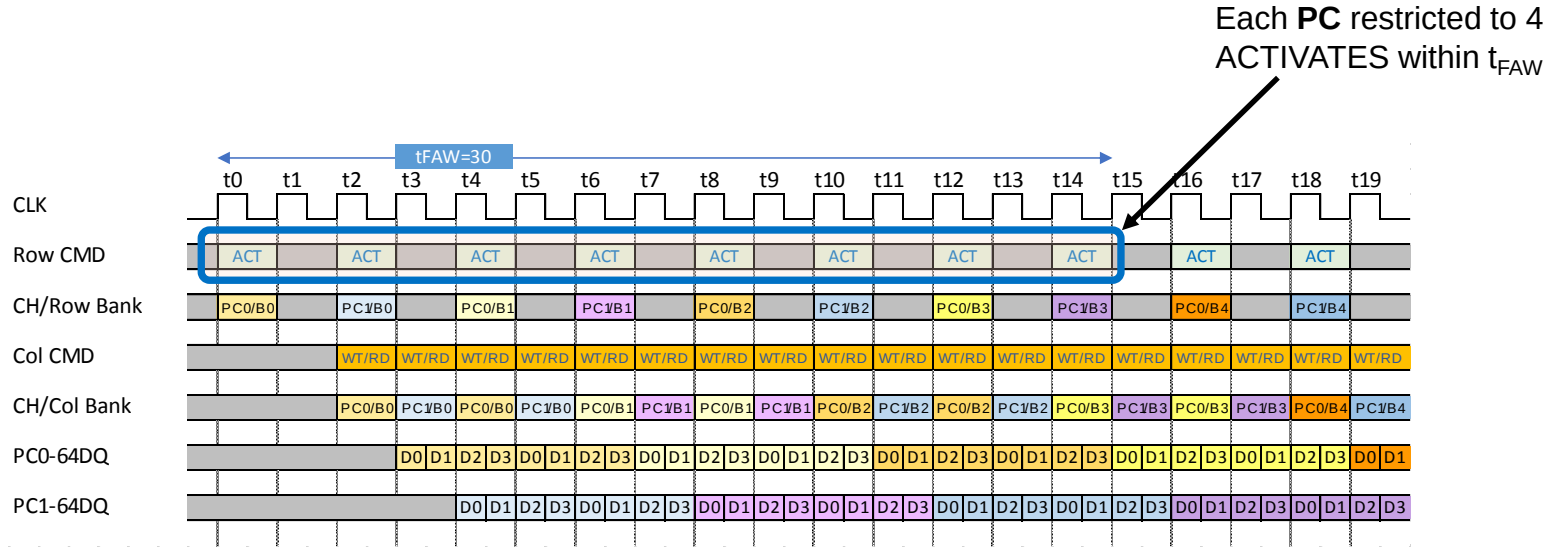
Compare: Legacy Mode

- Four active window (t_{FAW}) restriction
  - Restricts # of active banks to 4 within specified time periods
- Not supported by Intel® FPGA



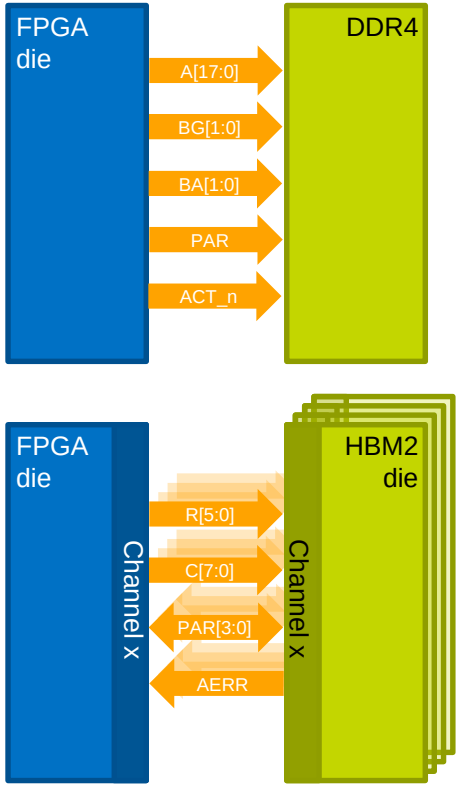
Benefit of Pseudo-Channel Mode

- t_{FAW} restriction is per pseudo-channel
- Separate pseudo-channels (**PC0**, **PC1**) can be activated



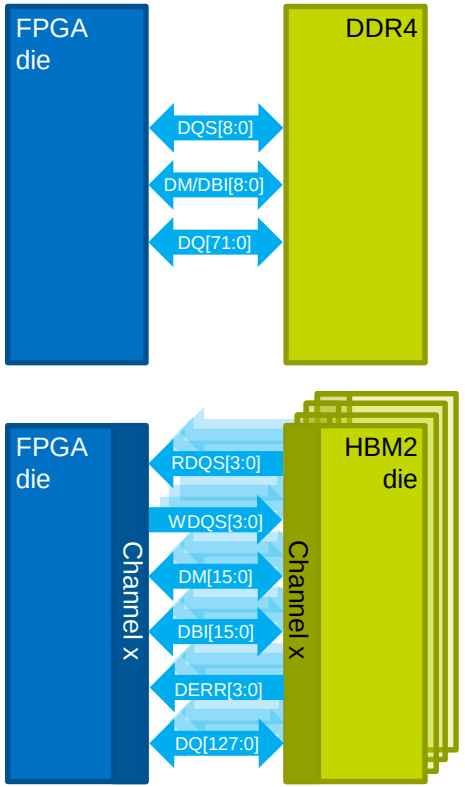
HBM2 Interface Signals: Address/Command

- DDR4
  - Address/command runs at single-data rate
  - Signals consist of:
    - 18-bit address (A)
    - 2-bit bank group (BG) and bank address (BA)
    - 1-bit parity (PAR)
    - 1-bit activate (ACT_n)
- HBM2
  - Address/command runs at double-data rate (DDR)
  - Signals consist of:
    - 6-bit row (R) and 8-bit column (C)
    - 4-bit parity (PAR)
    - 1-bit address error (AERR)



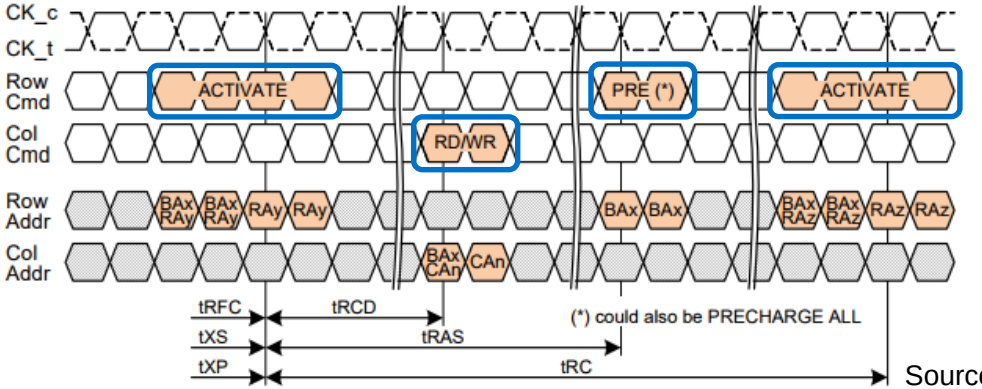
HBM2 Interface Signals: Data

- DDR4
  - Data strobe (DQS) is bidirectional
    - 1 DQS per 8 DQ
  - 1 data mask (DM) or data bus inversion (DBI) per 8 DQ
- HBM2
  - DQS is unidirectional
    - 1 RDQS and 1 WDQS per 32 DQ
  - 1 DM/DBI per 8 DQ
  - 1 data error (DERR) per 32 DQ



Timing

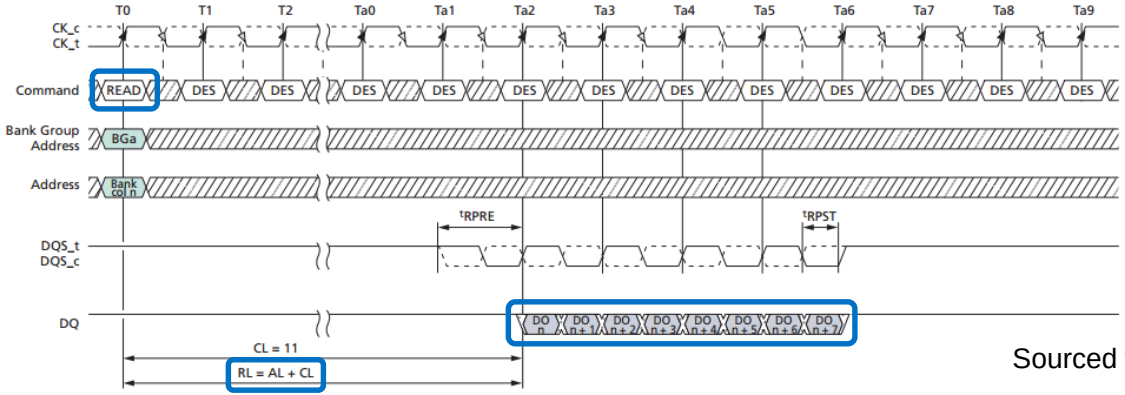
- Guaranteed within package
- Row activate commands = 2 cycles
  - All other commands = 1 cycle



Sourced from JEDEC

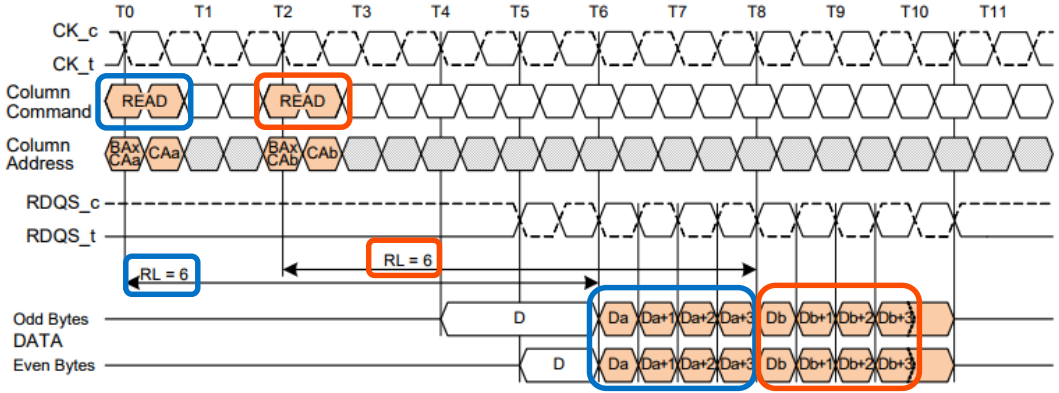
Read Timing

DDR4



Sourced from Micron

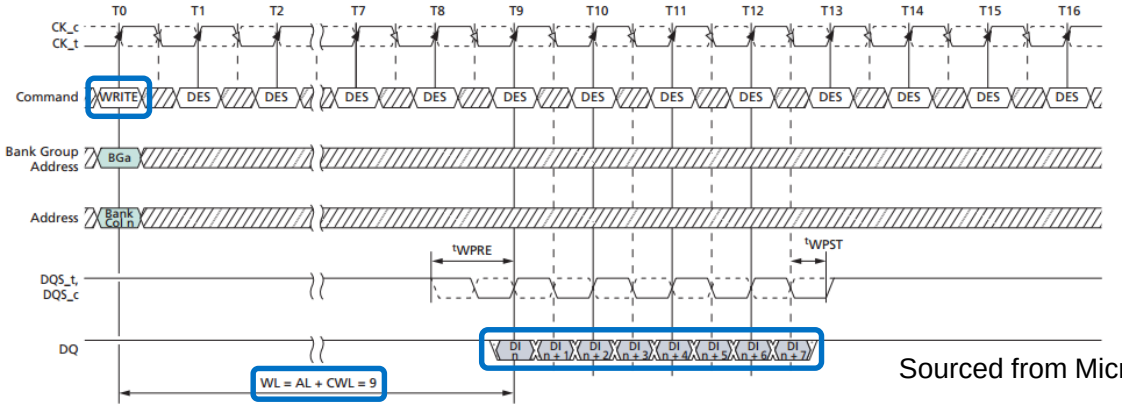
HBM2



Sourced from JEDEC

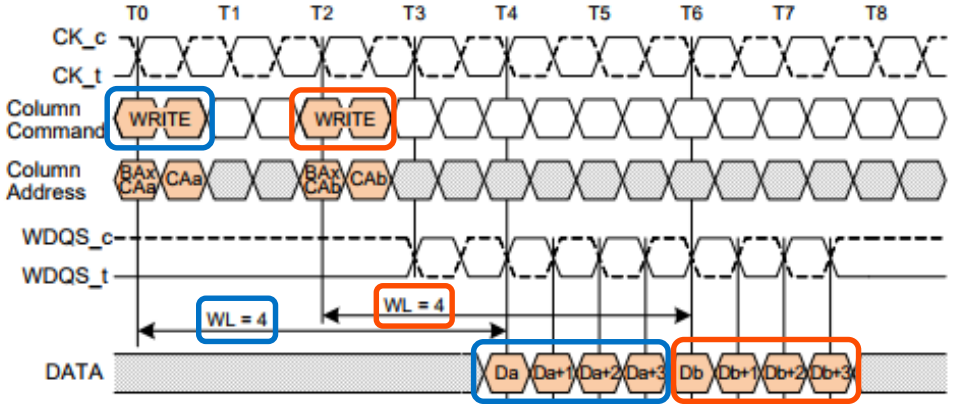
Write Timing

DDR4



Sourced from Micron

HBM2



Sourced from JEDEC

Additional HBM2 Features

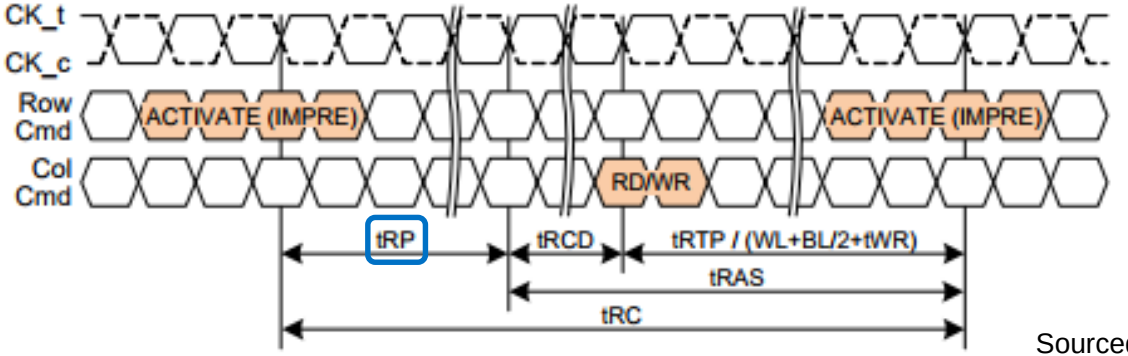
- Error Correction Code (ECC)
  - Cannot be enabled when data masking (DM) is enabled
- Temperature reporting
- Implicit precharge

Temperature Reporting

- Temperature compensated refresh
  - Reports refresh rate required by hottest device in the stack
  - **TEMP**[2:0]
- Catastrophic temperature sensor (**CATTRIP**)
  - Indicates if junction temperature exceeds catastrophic trip threshold (**CATTEMP**)
    - Programmed by manufacturer

Implicit Precharge

- Allows an ACTIVATE to be issued without a PRECHARGE
  - Applies to a different row in the same bank
- HMB2 PRECHARGE timings still apply



Sourced from JEDEC

Agenda

- Introduction to HBM2
- HBM2 architecture
- HBM2 controller (HBMC) features
- User interface to HBM2
- HBM2 IP implementation
- Performance and efficiency

High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

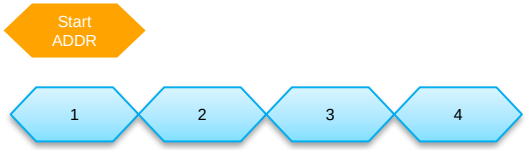
Controller Features

HBMC Features

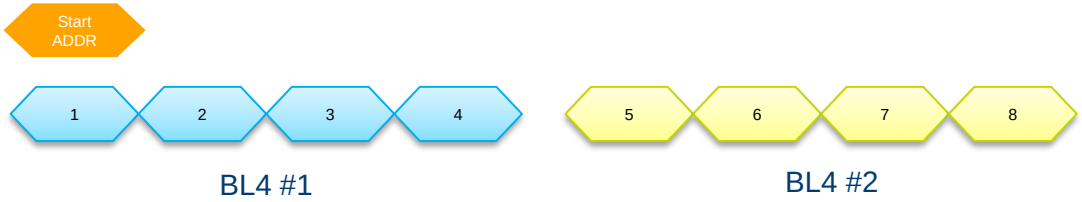
- Burst transactions
- Command scheduling
- Command priority
- Starvation limit for read commands
- Data re-ordering
- Address ordering schemes
- User REFRESH
- Self REFRESH
- User auto PRECHARGE
- Thermal control
- Thermal throttling
- Power down mode

HBMC Burst Options

- Default: BL4 transactions (32 bytes)



- Optional: pseudo-BL8 mode (64 bytes)
  - Two back-to-back BL4 transactions using a single start address



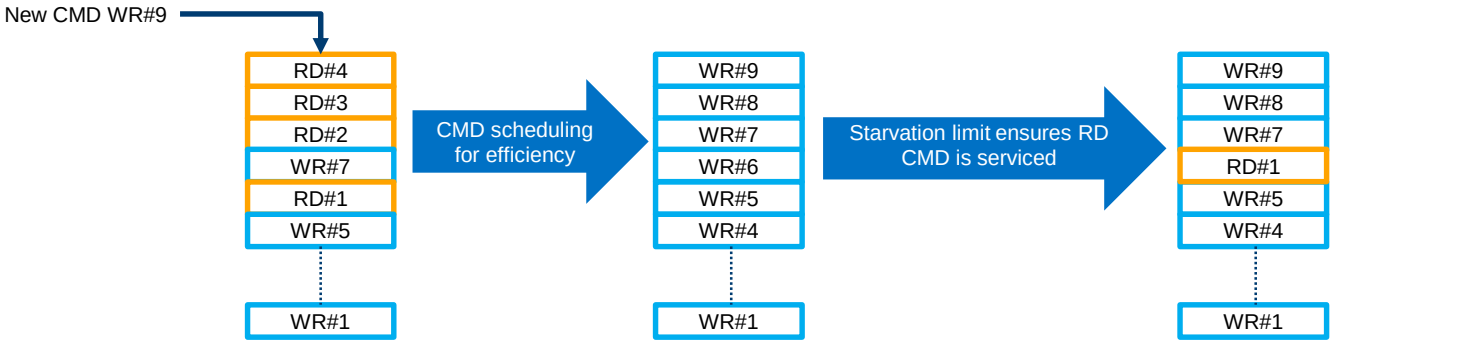
HBMC Command Scheduling and Priority

- Automatically (re)schedules commands to achieve maximum efficiency
- Conforms to Arm* AMBA* 4 AXI interface command ordering model

- Command priority can be set for read and write commands
  - Priority levels: NORMAL and HIGH
  - Set through Arm AMBA 4 AXI interface **AxQOS** (RD/WR quality of service) signal
- Commands with same priority levels serviced in a round-robin scheme

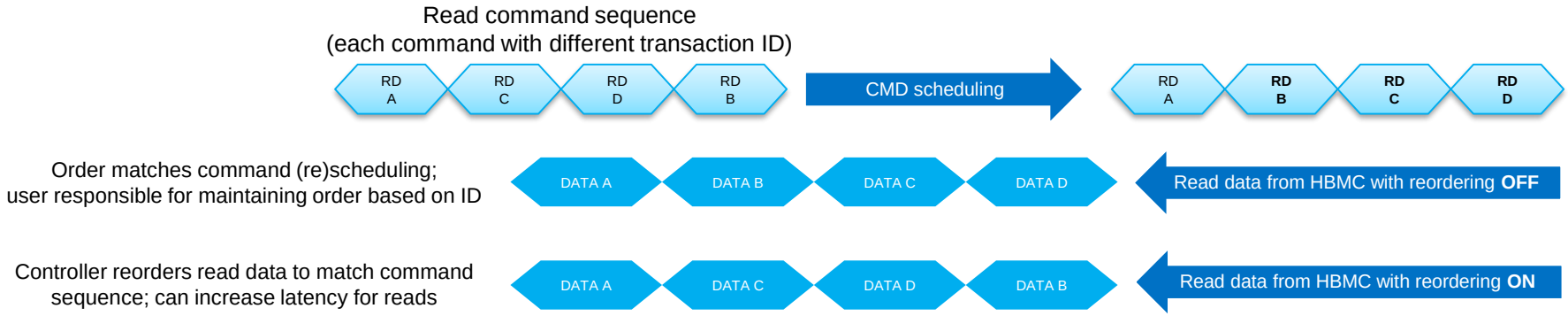
HBMC Read Command Starvation Limit

- Ensures that no READ command is left un-serviced
- Tracks the time every command has been waiting in the queue
- Issues the command if it reaches the starvation limit
- Limit is fixed in the controller (not adjustable)



HBMC Data Reordering

- Optionally reorder read data to match the order in which read requests are issued
- Controller performs reordering using reorder buffer (ROB) when different Arm* AMBA* 4 AXI interface transaction IDs are used
- When same transaction IDs are used, data must arrive in order
  - No reordering required



HBMC Addressing

- User provides address through the Arm* AMBA* 4 AXI interface
  - 28-bit (4 GB) or 29-bit (8 GB) read and write address buses
  - Includes row address [13:0], column address [5:0], bank address [3:0], and stack ID (SID) (8 GB devices only)
- Address configurations change based on
  - Burst size: BL4 vs pseudo-BL8
    - Col[0] tied to 0 for BL4
    - Col[1:0] tied to 0 for BL8
  - Device density
    - SID available only in 8H/8GB device

Address Ordering

- Select address ordering scheme based on user access pattern
  - Order in which the controller interprets the incoming read and write addresses from the user logic
- Supported address ordering configurations
  - SID-BG-BA-ROW-COL (bank group (BG) = BA[3:2])
  - SID-ROW-BA-COL-BG (default)
    - Helps to use “shorter” DRAM access timing between bank groups
  - ROW-SID-BA-COL-BG

HBMC Refresh Options

- Supports user or controller managed refresh
- Direct user control *(supported in version 18.0 and later)*
  - Refresh all or per-bank
  - Refresh commands issued through sideband Arm* AMBA* APB interface
- Controller managed refresh policies
  - Default flexible refresh: let controller decide optimal refresh time
  - “Pre-pay”: controller issues refreshes earlier when controller is idle
  - “Post-pay”: controller postpones refreshes when controller is busy

Auto Precharge Policies

- **HINT** (default)
  - User issues request for precharge on Arm* AMBA* 4 AXI interface **AxUSER** sideband signals
  - Controller decides when to issue
- **FORCED**
  - User issues request for precharge
  - Precharge request immediately enforced
- Policies can be different for reads and writes

Thermal Control & Throttling

- Control
  - HBMC adjusts refresh rate for hottest device in stack based on **TEMP** signal
  - HBMC halts all traffic to the stack if **CATTRIP** goes active (**CATTEMP** exceeded)
- Throttling
  - Helps control thermal runaway by throttling input commands
  - User defined junction temperature setting and throttle ratio (throttling frequency)
  - Controller de-asserts **AxREADY** and **WREADY** signals when actively throttling

Power Down Mode

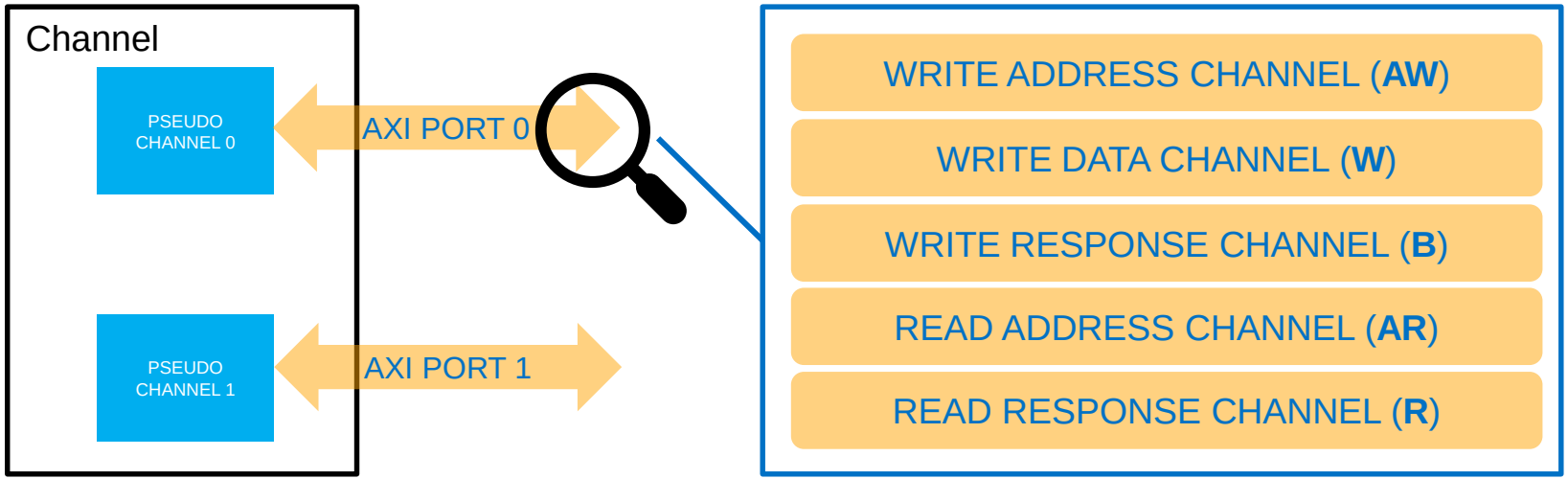
- Controller puts HBM2 into power-down state when idle for long period of time
- Each channel can enter power down mode individually

High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

User Interface to HBMC

Arm* AMBA* 4 AXI Interface Ports and Channels

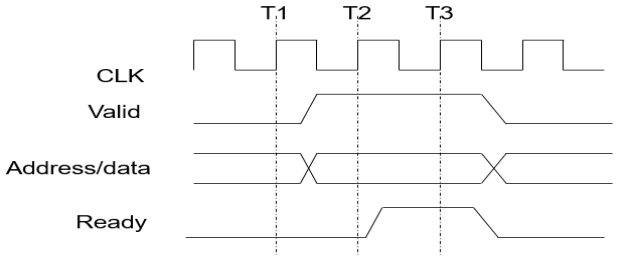
- Two Arm AMBA 4 AXI ports per HBM2 channel
  - One for each pseudo-channel handles all reads, writes, and responses
- Individual signals named after HBM2 channel, pseudo-channel, and role
  - Ex.: **axi_0_1_awid** is the write address transaction ID on channel 0, pseudo-channel 1



Arm* AMBA* 4 AXI Standard Handshaking

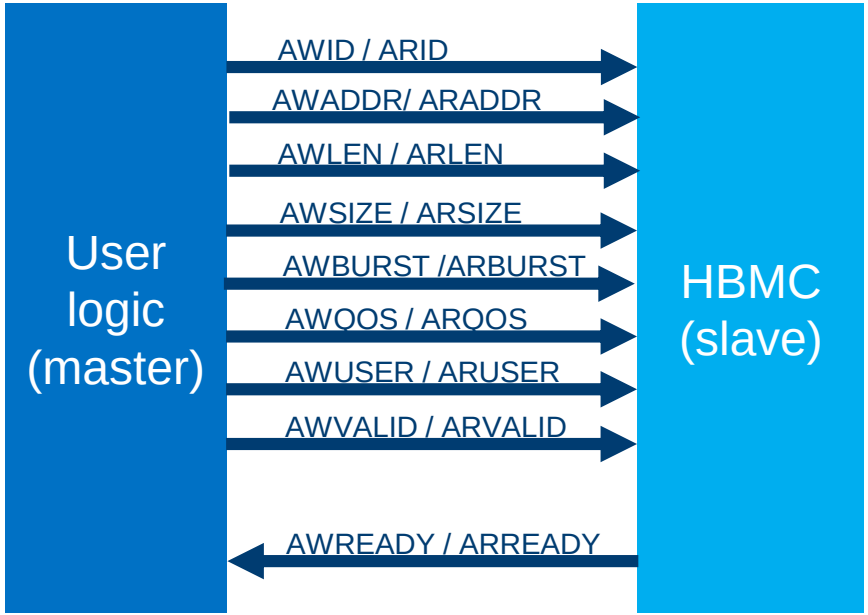
The Arm AMBA 4 AXI interface handshaking protocol is used on all 5 channels on all ports

- Uses **VALID/READY** handshake process
- Source (master in **AW**, **AR**, **W**; slave in **B**, **R** channels) generates **VALID** signal
- Destination (slave in **AW**, **AR**, **W**; master in **B**, **R** channels) generates **READY** to indicate it can accept data
- Transfer occurs only when both **VALID** and **READY** are high



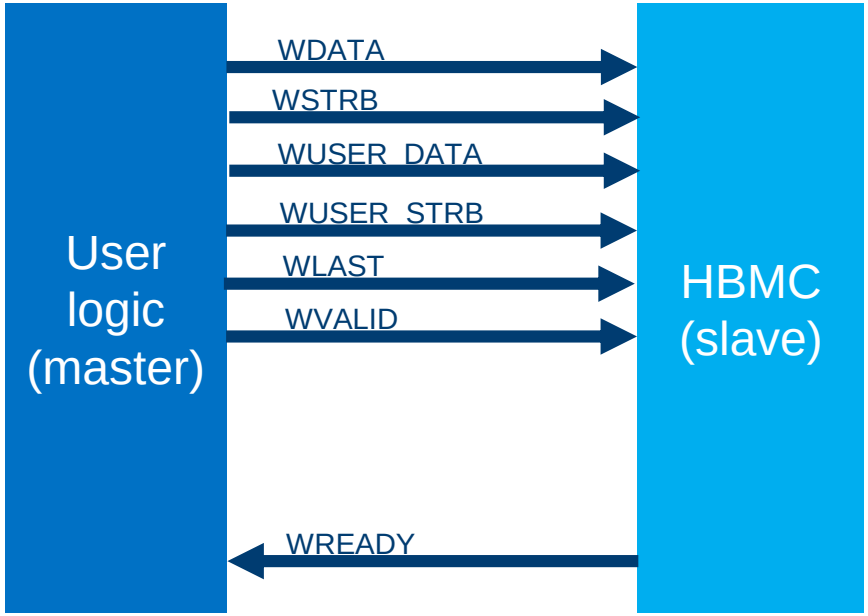
AW and **AR** Channel Signaling

- **ID**: transaction ID
- **ADDR**: address
- **LEN**: # of transfers in a burst; **LEN** = 0 since only 1 burst transfer (BL4 or 8) supported at a time
- **SIZE**: burst size (32 (BL4) or 64 (BL8) bytes)
- **BURST**: burst type (not supported)
- **QOS**: high or normal priority
- **USER**: user auto-precharge control
- **VALID**: address valid
- **READY**: address ready



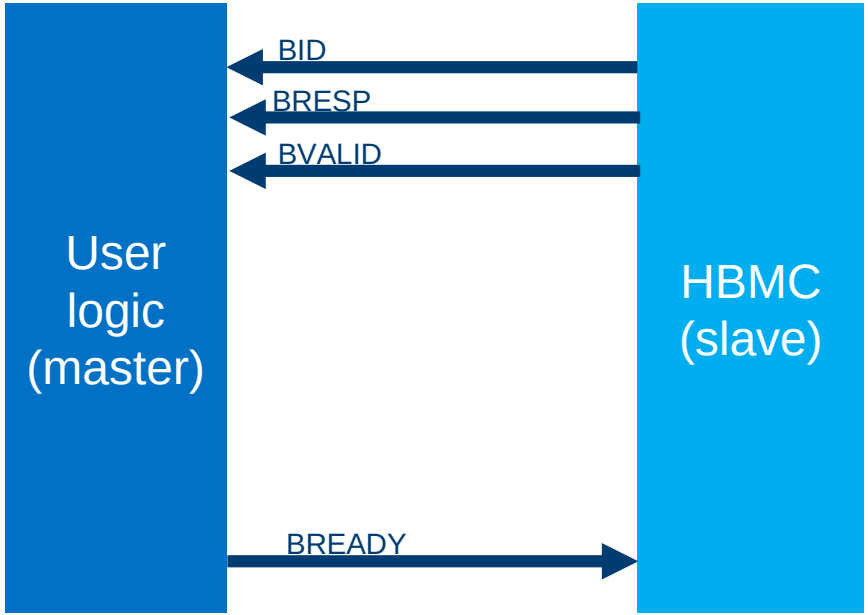
W Data Channel Signaling

- **DATA**: write data (256 bits)
- **STRB**: write strobe (byte enables)
- **USER_DATA**: extra data for ECC
- **USER_STRB**: ECC data byte enables
- **LAST**: last transfer in write burst
- **VALID**: write valid
- **READY**: write ready



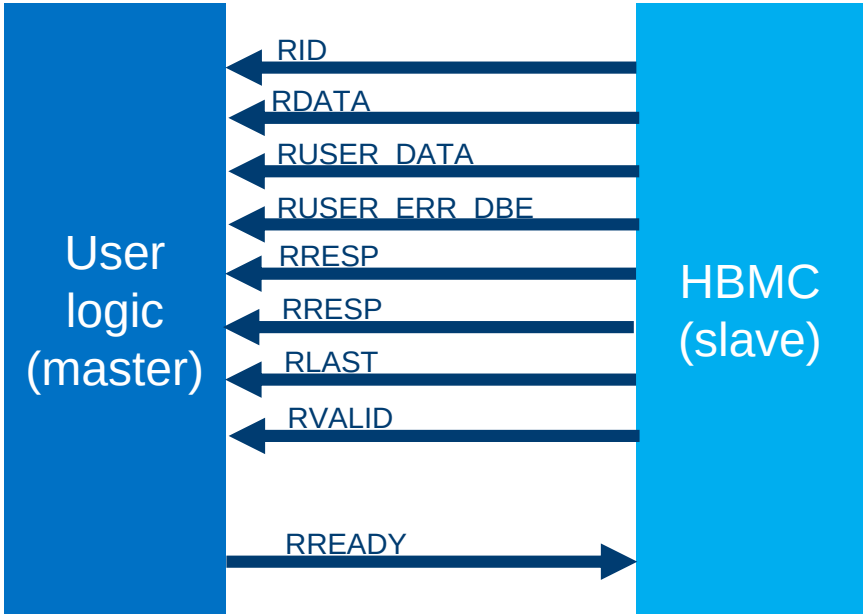
B Channel Signaling

- **ID**: response transaction ID
- **RESP**: write response
  - OKAY, exclusive OK (not supported), slave error, decode error
  - HBMC should always return OKAY
- **VALID**: write response valid
- **READY**: write response ready



R Channel Signaling

- **ID**: read transaction ID
- **DATA**: read data (256 bits)
- **USER_DATA**: extra data for ECC
- **USER_ERR_DBE**: double bit error
- **RESP**: read response
  - OKAY, exclusive OK (not supported), slave error, decode error
  - HBMC should always return OKAY
- **LAST**: last transfer in read burst
- **VALID**: read valid
- **READY**: read ready



Agenda

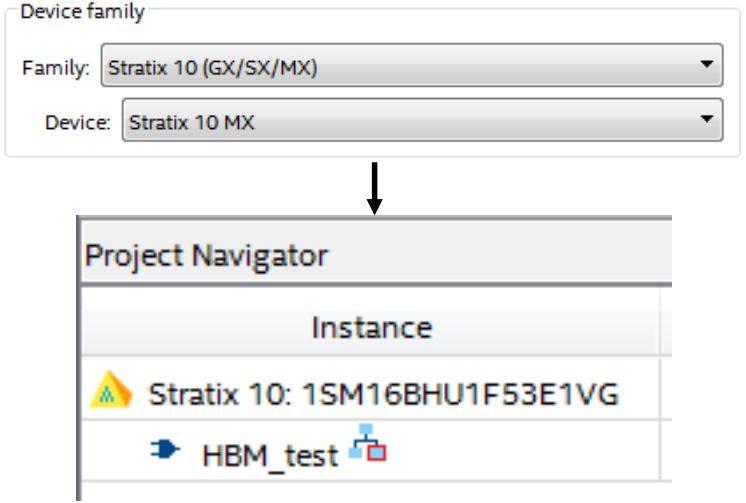
- Introduction to HBM2
- HBM2 architecture
- HBM2 controller (HBMC) features
- User interface to HBM2
- HBM2 IP implementation
- Performance and efficiency

High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

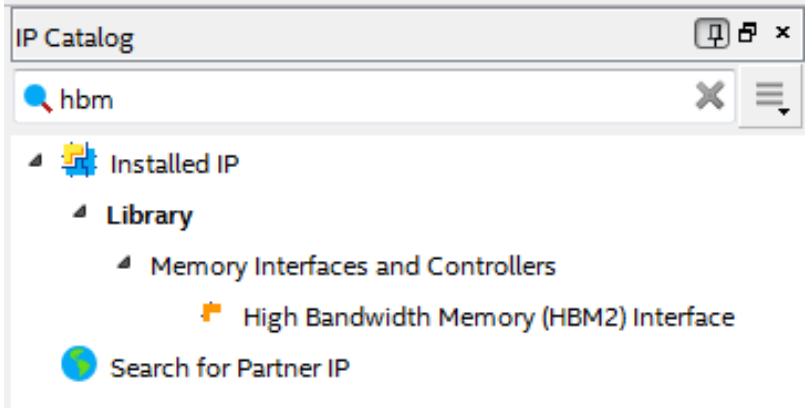
HBM2 Implementation

Getting Started

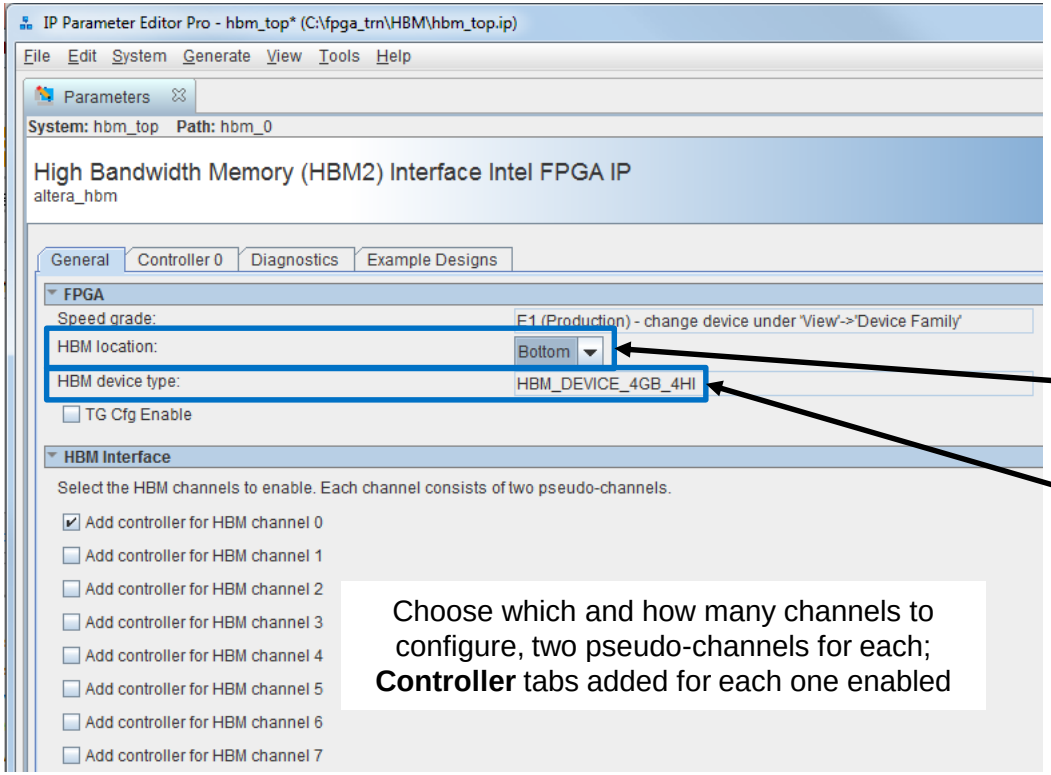
HBM2 can only be added to a project targeting an Intel® Stratix® 10 MX device in the Pro edition of the Intel Quartus® Prime software v. 17.1 or later



IP Catalog in the Intel Quartus Prime software or Platform Designer



General Tab



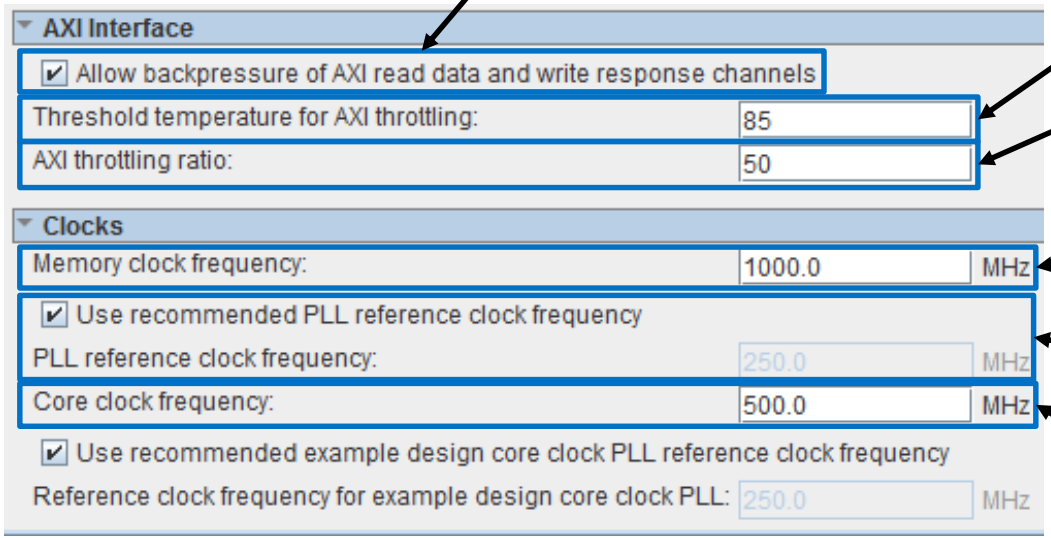
Choose which and how many channels to configure, two pseudo-channels for each; **Controller** tabs added for each one enabled

Configure the HBMC for the UIB at either the top or bottom of the device

Indicates whether selected device's stack is 4 high or 8 high

General Tab (cont.)

Enable the Arm* AMBA* 4 AXI interface soft logic adapter



Temperature threshold for throttling
% of transactions where controller purposefully de-asserts READY signals past temperature threshold
0 = no throttling, 100 = full % of transactions where controller purposefully de-asserts READY signals past temperature threshold

Memory clock frequency based on selected device speed grade

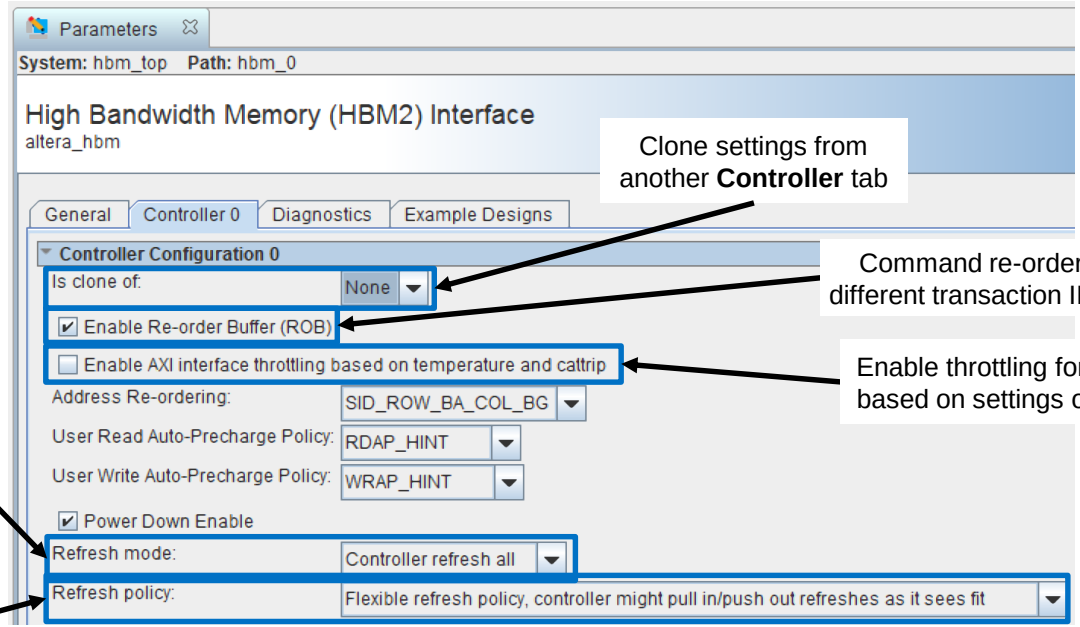
PLL reference clock frequency to supply to the IP

Speed of core logic based on device speed grade and ability to close timing

HBM2 Supported Frequencies

	Intel® Stratix® 10 MX device speed grade		
	-1	-2	-3
HBM2 interface maximum frequency	1000 MHz	800 MHz	600 MHz

Controller Tab



Clone settings from another **Controller** tab

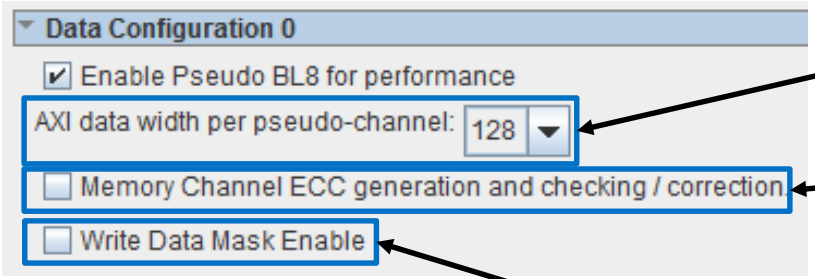
Command re-ordering when different transaction IDs are used

Enable throttling for this controller based on settings on **General** tab

Controller or user refreshes all banks or user refreshes specific banks; controlled by Arm* AMBA* APB sideband

Controller refresh policy: flexible, pre-pay, or post-pay

Controller Tab (cont.)



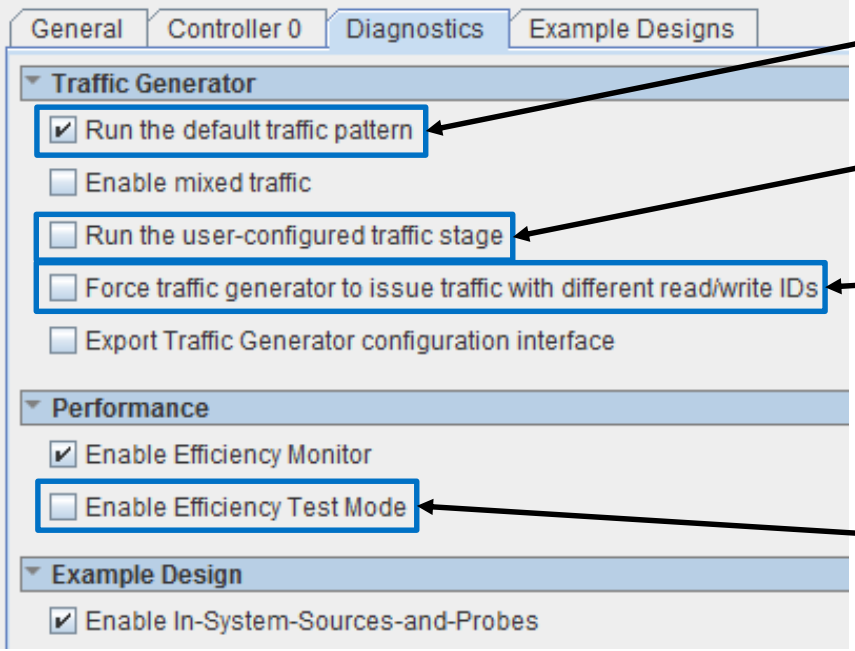
Data width per pseudo channel of 128 bits;
144 bits option if ECC and DM not being used

Single-bit error correction, double-bit
error detection if DM not being used

Enable write data masking if
ECC not being used

Diagnostics Tab

Choose diagnostic options for generated example design *(discussed more in next section)*



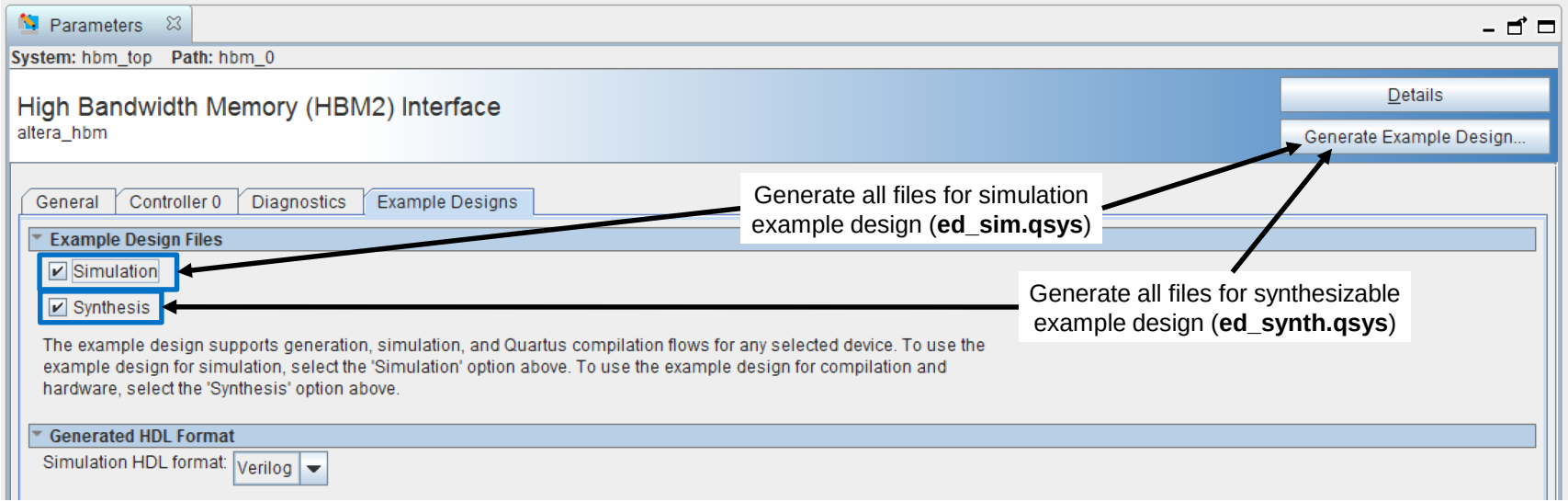
Synthesizable traffic generator (TG) performs combination of single, block, random write/read transactions

Create custom traffic through EMIF toolkit or Avalon® memory-mapped slave interface of TG

Higher efficiency, but data mismatches possible if reorder buffer disabled

Higher efficiency by having TG not check that read data matches written data

Example Designs Tab



- If **Simulation** and/or **Synthesis** options disabled when clicking **Generate Example Design**, **.qsys** file created along with script to build rest of design later

High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices

HBM2 Performance and Efficiency

Calculating HBM2 Bandwidth

Maximum and effective bandwidth can be calculated for the interface

- Maximum bandwidth
  - Per channel
    - 128 bits DDR data, operating at 1 GHz
  - Maximum bandwidth (for all 8 channels)
    - = 32 GBps * 8 = 256 GBps
- Effective bandwidth
  - Maximum bandwidth * controller efficiency
  - 256 GBps * 0.9 = 230.4 GBps (90% efficient)

What is Controller Efficiency?

Controller efficiency is a measure of % utilization of the memory DQ bus

- Why do we care about efficiency?
  - Good indicator of how efficient write and read commands are converted to HBM2 memory commands
  - More transactions can be scheduled in a highly efficient controller in a given time
  - Efficient command scheduling provides high efficiency

Factors That Affect Controller Efficiency: HBM2

- Memory accesses
  - Sequential memory addressing provides higher efficiency vs. random addressing
    - Frequent PRECHARGE and ACTIVATE commands needed for random transactions affect efficiency
  - Transactions that require frequent bus turnaround (R to W, W to R)
    - Concurrent W/R supported at Arm* AMBA* AXI interface to help mitigate this
  - Use of Bank Groups (BG = BA[3:2])
    - “Short” access timing requirements used when accessing different BG
    - ”Long” access timing requirements used for accessing same BG
- Memory refresh policy
  - Frequent refreshes vs. refreshes postponed until absolutely necessary
- Precharge policy
  - Auto precharge helps with random transactions

Factors That Affect Controller Efficiency: HBMC

- Transaction burst length
  - Pseudo-BL8 transactions vs BL4
  - Different bank groups used for pseudo-BL8 accesses
- Command scheduling
  - Efficient command scheduling performed when different transaction IDs used
- Re-order buffer
  - If enabled, can add latency when re-ordering read data
- Core clock frequency vs. HBM2 clock frequency
  - Operate the core as close as possible to HBM2 frequency
  - However, faster core frequency means tougher timing closure

Measuring Controller Efficiency

When enabled, the Efficiency Monitor calculates efficiency when performing a functional simulation

- Measured at user Arm* AMBA* 4 AXI interface
  - Recall: one interface per pseudo-channel (PC)
- Controller issuing READY indicates how quickly user transactions processed
  - Separate READY for **AW**, **AR**, and **W** channels (where controller is destination)
- Calculate ratio of total # of accepted data transactions at interface to total transaction time
  - Concurrent WR/RD transactions supported by interface
- Efficiency scaled by core vs. HBM2 clock ratio

HBM2 Read Latency

Read latency is the total round trip delay from user logic issuing a read command to valid read data available at the Arm* AMBA* 4 AXI interface

- Read latency includes
  - Command latency from the user interface to the HBM2 through the HBMC
  - Memory read latency
  - Delay of read data from HBM2 through HBMC until available at the user read data port
- Efficiency monitor displays the minimum read latency

Simulating for High Efficiency

Configure the IP with the settings listed here to simulate for high efficiency

- **General** tab
  - Enable the Arm* AMBA* 4 AXI interface adapter
  - Set memory clock frequency to **1000 MHz**, core clock to **720 MHz** (for speed grade -1 device)
  - Enable **Use recommended example design core clock PLL reference…**
- **Controller** tab
  - Disable the re-order buffer
  - Address re-ordering: **SID_ROW_BA_COL_BG**
  - Auto-precharge policy for reads and writes: **xAP_HINT**
  - **Controller refresh all** and **Flexible refresh policy**
  - Enable **Pseudo BL8**

Simulating for High Efficiency (cont.)

- **Diagnostics** tab
  - Enable **Run the default traffic pattern**
  - Enable the TG to **issue traffic with different read/write IDs**
    - With re-order buffer disabled, data mismatches may occur, but can be ignored for this sim
  - Enable the **Efficiency Monitor** and the **Efficiency Test Mode**
    - In **Efficiency Test Mode**, data validation not performed so transactions can be sent sooner
    - More data mismatches may occur

End of Part 3

To continue the training, check out:

- Part 1: High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices: Introduction & Architecture
  - *https://www.altera.com/support/training/course/ohbms10.html*
- Part 2: High Bandwidth Memory (HBM2) in Intel® FPGA Stratix® 10 MX Devices: HBMC Features & User Interface
  - *https://www.altera.com/support/training/course/ohbmcs10.html*

Legal Disclaimers/Acknowledgements

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel, the Intel logo, Intel Inside, the Intel Inside logo, MAX, Stratix, Cyclone, Arria, Quartus, HyperFlex, Intel Atom, Intel Xeon and Enpirion are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

OpenCL is the trademark of Apple Inc. used by permission by Khronos

*Other names and brands may be claimed as the property of others


intel®
experience
what's inside™