

ARCHITECTING A HIGH PERFORMANCE DDR3 DRAM CONTROLLER

Sajish Sajayan

S3Craft Technologies

25th March, 2015

AGENDA

- ▣ DDR Controller Architectural Considerations
 - Performance
 - Flexibility
- ▣ DDR DRAM Architecture, Inefficiencies
- ▣ Overcoming DDR DRAM Limitations
- ▣ A High Performance DDR3 DRAM Controller
- ▣ Summary

PERFORMANCE CONSIDERATIONS

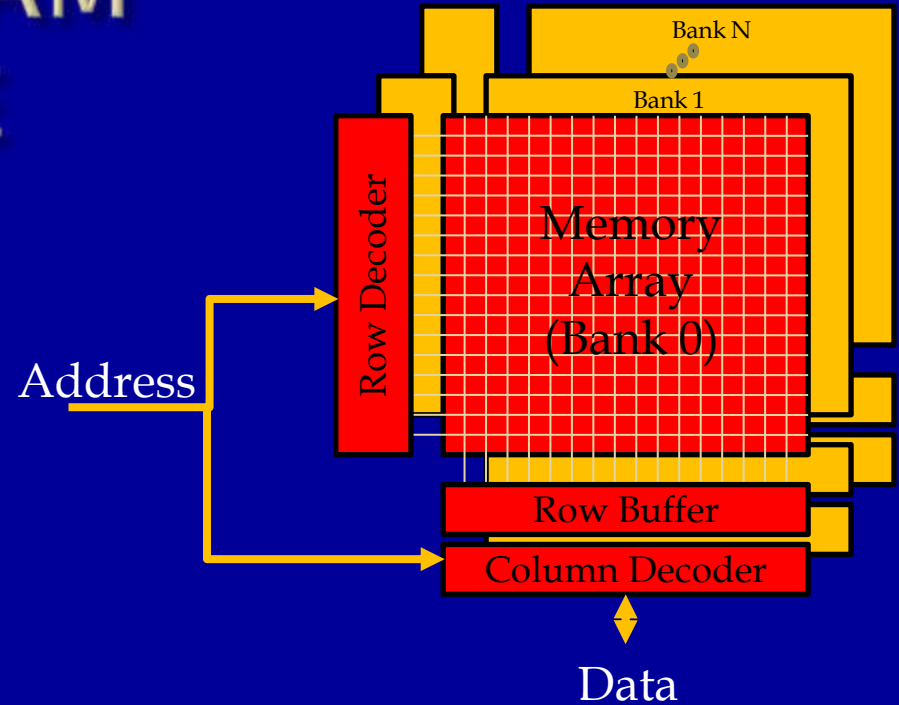
- ▣ Bandwidth
 - Increased efficiency of DDR accesses results in
 - More bandwidth from a particular speed-grade, or,
 - Sufficient bandwidth from a lower speed-grade
- ▣ Latency
 - Reduced latency results in
 - Less processor/accelerator idle time, and therefore,
 - Reduced caching requirements
 - Latency issues exacerbated in multi-core systems
- ▣ Frequency
 - Increased frequency for increased bandwidth
- ▣ Area
- ▣ Area vs performance
 - More performance usually needs more area, due to parallelization of operations

FLEXIBILITY CONSIDERATIONS

- ▣ Compile time configurability
 - Core-side protocol : AXI3, AXI4, others
 - Number of slave (core-side) ports
 - Number of supported Ranks
 - Depth of internal data buffers
 - DRAM data width
 - Core-side address and data width
 - Synchronous vs Asynchronous interfaces
 - Frequency dependent pipelining
- ▣ Run-time configurability
 - Memory speed-grade support
 - Priority scheme for access requests
 - Mapping of Core-side address to DDR rank/bank/row/column
 - Number of supported Ranks
 - DRAM Data width

A NOTE ON DRAM ARCHITECTURE

- Structure
 - 3-dimensional
 - Bank, row, column
- Access protocol
 - Activate (Row), Access (Column/s), Precharge (Row)
 - Per bank, only one row active at a time
- Resource Constraints
 - Bank
 - Address/Command lines
 - Data lines



DRAM INEFFICIENCIES...

- ▣ Activate → n *Access → Precharge sequence requirement
 - And timing constraints between these commands
 - ▣ For every random read or write request
- ▣ Only 1 row per bank can be active at a time
 - Accessing another row in same bank mandates closing of previously open row in the bank
 - Back to back accesses to different rows in the same bank severely impact bus efficiency
- ▣ Read to write, and write to read turnaround time
- ▣ Refresh time
- ▣ Shared command and data lines

... AND OVERCOMING THEM (1)

- ▣ Method
 - Queue up the incoming access requests
 - Look ahead into the command queue, and group together accesses
 - ▣ to the same bank
 - ▣ to the same row within a bank
 - Prioritize accesses to the current open row within the bank
 - ▣ For each bank
 - ▣ Club together the reads, and the writes, prevent interleaving of read and write requests
 - Interleave accesses among the banks
 - ▣ Hide the activate / precharge latency for a bank behind access commands for other banks
 - Optionally, decide on the fly to precharge (or not) at the end of access

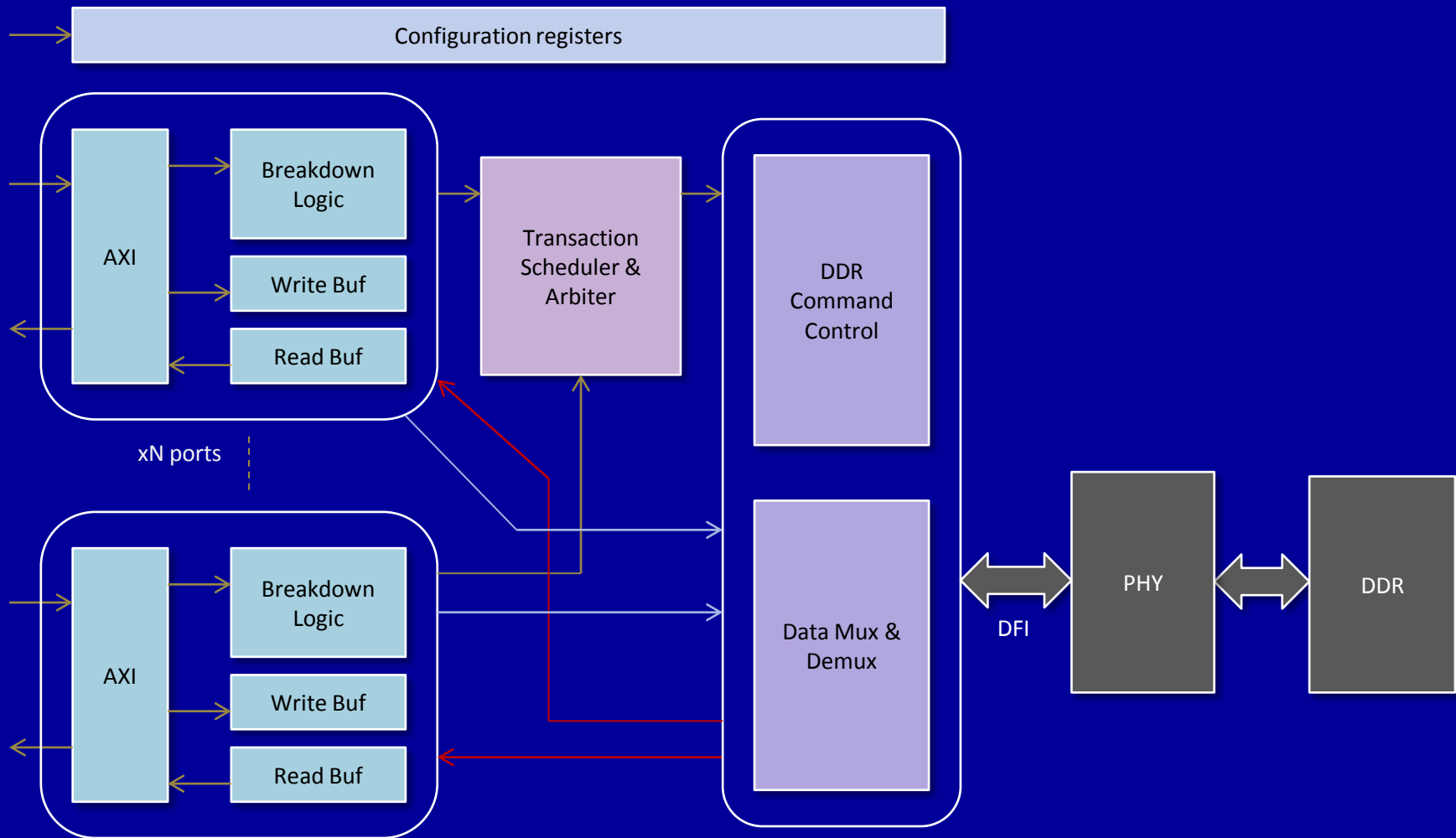
... AND OVERCOMING THEM (2)

- ▣ Care-about
 - Ensure timing constraints are met
 - ▣ Between the different commands
 - ▣ Across banks
 - ▣ Across ranks
 - Ensure sufficient refresh cycles
 - Provide hooks to enable high priority requests to bypass the regular reordering algorithm
 - Prevent starvation
 - Ensure read data is re-ordered in the expected manner, when sent back to the core-side requestor

AN ILLUSTRATION (BANK INTERLEAVING)

	Cycle	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	25	26
w/o Look Ahead	Commands	A C T 0				R D 0				R D 0				P R E 0	A C T 1				R D 1									
	Data										D0	D0	D0	D0	D0	D0	D0	D0						D1	D1	D1	D1	
w/ Look Ahead	Commands	A C T 0	A C T 1		R D 0				R D 0				P R D 1	A R E 0														
	Data										D0	D0	D0	D0	D0	D0	D0	D0	D1	D1	D1	D1						

DDR3 Memory Controller



DDR3 Controller – AXI SLP

□ AXI Slave Port

■ AXI3/AXI4

- Supports narrow transfers
- Supports wrap
- Parametrizable Addr & Data widths

■ Flexible breakdown logic

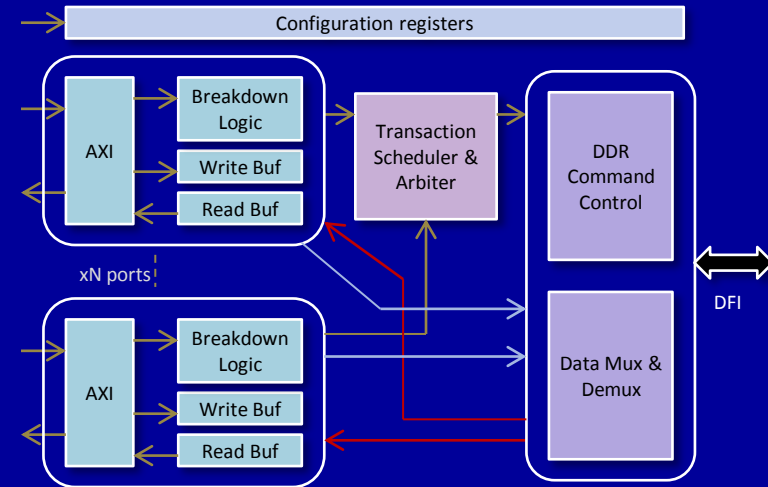
- Breaks down AXI transactions into DDR transactions
- Programmable mapping of AXI address to DDR rank, bank, row, column
- Supports different DDR data widths

■ Parametrizable buffer depths

- For read/write buffers (separately)

■ Optional read re-order buffer

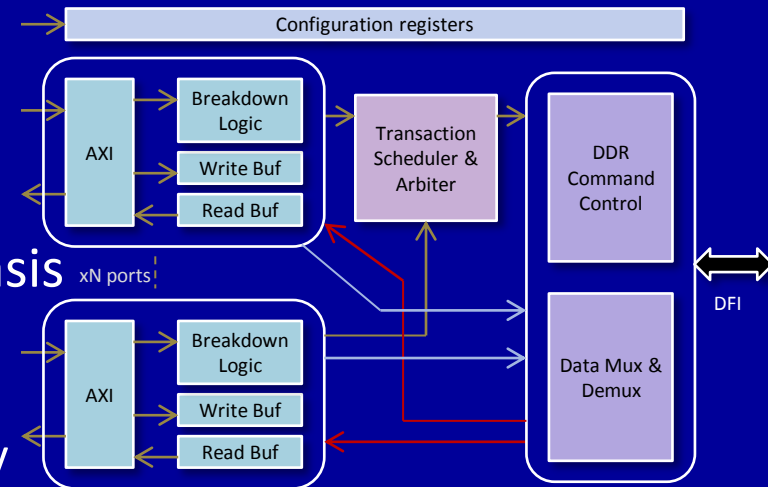
- Supporting AXI tags, data interleaving



DDR3 Controller – TRS

Transaction Scheduler

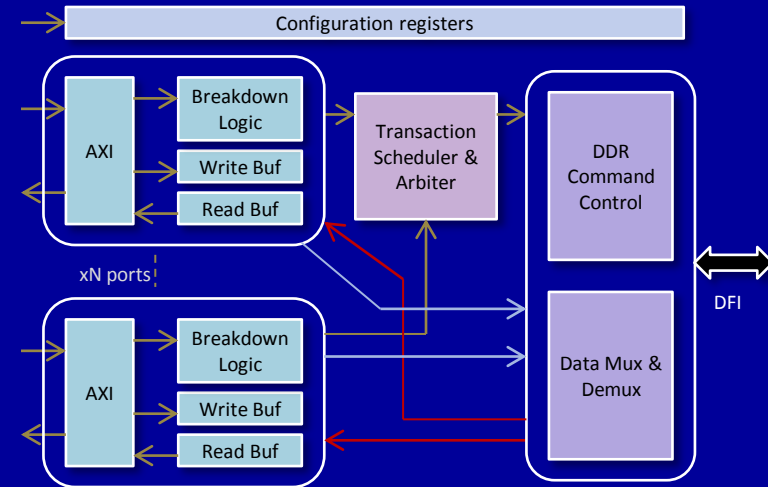
- Prioritizes requests
 - Based on QoS / port (programmable)
 - 2 Read Queues, 1 Write Queue
- Re-orders requests, on per-bank basis
 - Prioritizes requests to open row
 - Reads over writes
 - High priority read over normal priority
- Ensures no starvation
 - Promotion of normal priority request to high priority
 - Promotion of write requests based on thresholding
- Parametrizable request buffer depths
 - Larger depth for more optimal request selection
- Write Caching & Read Bypass



DDR3 Controller – CMS

▣ Command Scheduler

- Ensures all DDR timing reqs met
 - ▣ Between different commands
 - ▣ Between banks
 - ▣ Between ranks
- Arbitrates among banks
 - ▣ High/Normal priority reads, writes
- Arbitrates among ranks
- DFI interface
- Leveling
 - ▣ Both master and slave configurations
- Supports different speed grades of DDR3 memories



DDR3 Controller – Summary

▣ Standards Compliance

- Complies to DDR3/3L specification
- Supports AXI4 & AXI3 access ports
- PHY interface compliant to DFI 3.1
- Supports 1:2 clocking
- APB compliant configuration port

▣ Architecture Highlights

- 3,200 MT/s @ 28nm (HP) & 1,600 @ 65nm (LP)
- Low latency paths to & from PHY
- Smart command re-ordering scheduler
- 2 read queues with different priorities
- Tagged AXI: Out-of-order read return data
- Support for multiple memory ranks

▣ Quality & Performance

- Configurable QoS/port based arbitration
- Built-in anti-starvation mechanisms
- Power management for controller & DRAM
- Verified with established DDR3 PHY & VIP

Q & A