



Achieving High Memory Bandwidth for Cortex Family-based Video SoCs Using Multiple DRAM Channels

Drew Wingard
Sonics, Inc.



Introduction

- What are the applications for High Quality High Definition (HQHD) SoCs?
- Key characteristic: relentless push for higher quality video experiences

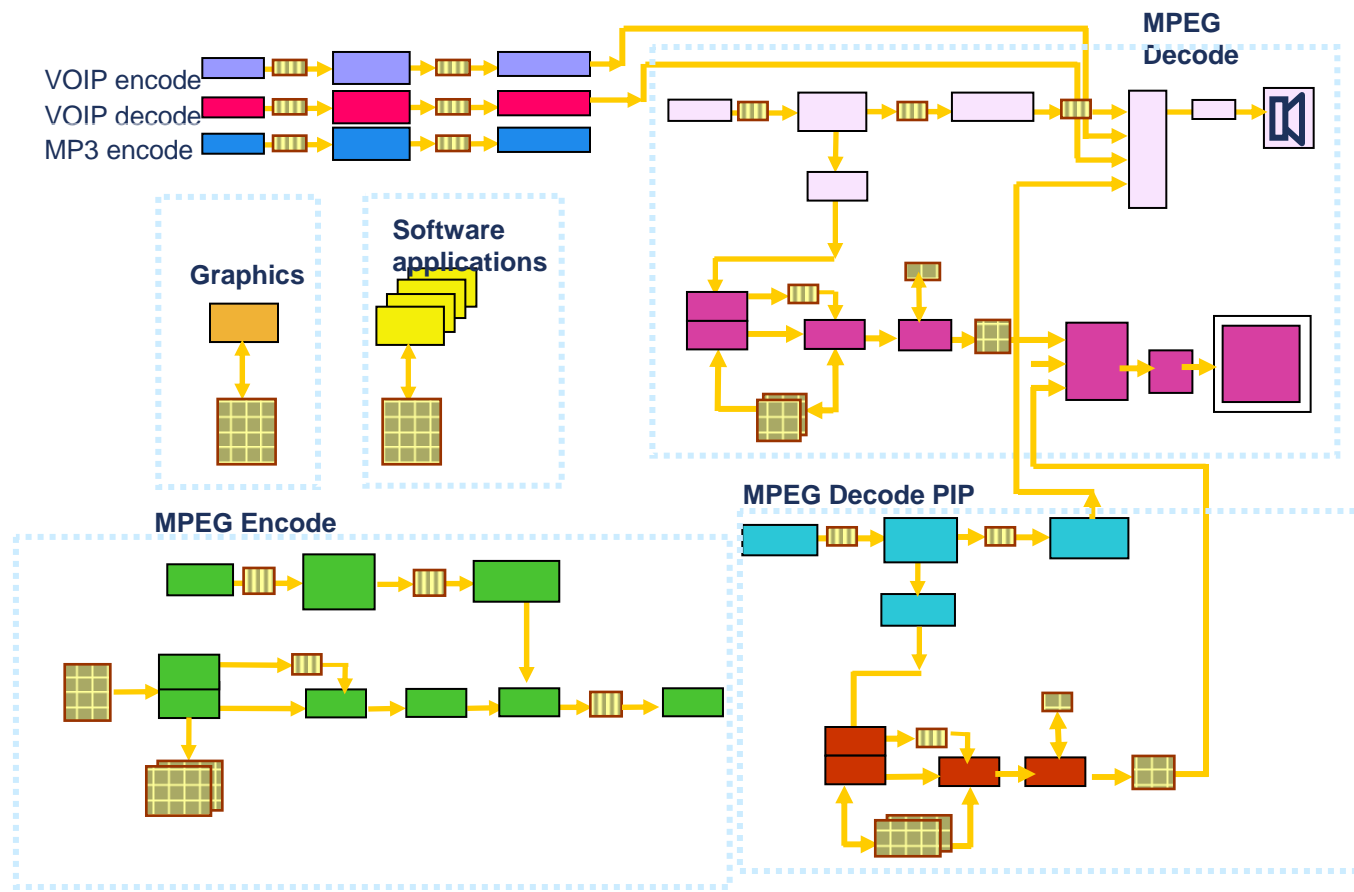


Agenda

- ***HQHD video SoCs***
- High-bandwidth DRAMs
- Delivering high DRAM bandwidth in video SoCs
- Experimental results

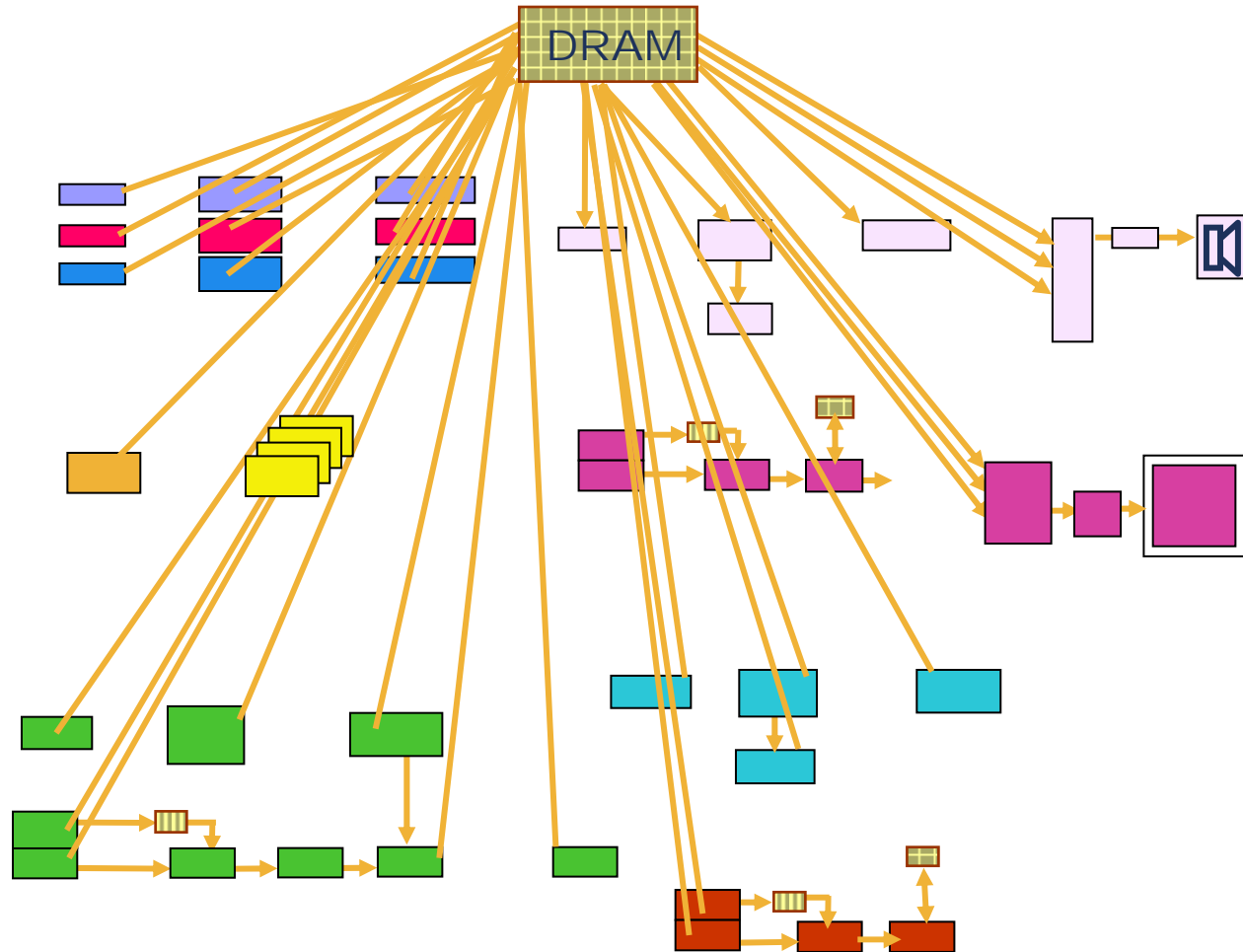
Concurrency in Video SoCs

- Video SoCs process lots of data in parallel, but communicate...



Concurrency in Video SoCs

- Video SoCs process lots of data in parallel, but communicate...



Concurrency in Video SoCs

- Video SoCs process lots of data in parallel, but communicate...
- Assertion: video SoC applications have \gg 50% of system traffic to/from external DRAM
- Consumer volumes and price points demand cheapest DRAM configurations that support required performance
- Implications:
 - SoC architecture is mostly a fan-in tree to external DRAM
 - Maximizing delivered DRAM **throughput** and **efficiency** are key

Video SoC Traffic and Requirements

- DRAM traffic characteristics
 - Long burst: streaming I/O, natural motion, audio
 - Short burst: cache misses, graphics
 - 2D: video decoding
- Performance requirements
 - High bandwidth @ high efficiency (max perf. @ min. cost)
 - Predictable throughput @ bounded latency (soft real time)
 - Low latency for bursty CPU traffic
 - Increased application software load needs faster CPUs, such as ARM Cortex-A series

HQHD Video DRAM Bandwidth Requirements

Example: External DRAM Bandwidth Increases for HQ HDTV

Decoding Formats	Motion Compensation Image Enhancement	Bandwidth (GB/s)	
MPEG-2 HD		1.35	HD
MPEG-2 HD	Motion Compensation Judder removal	2.6	HD
H.264	High Definition Natural Motion	4.6	HD
H.264	HDNM - 2009	5.2	HQHD
Dual H.264	HDNM - 2010	10.5	HQHD

Source: Philips

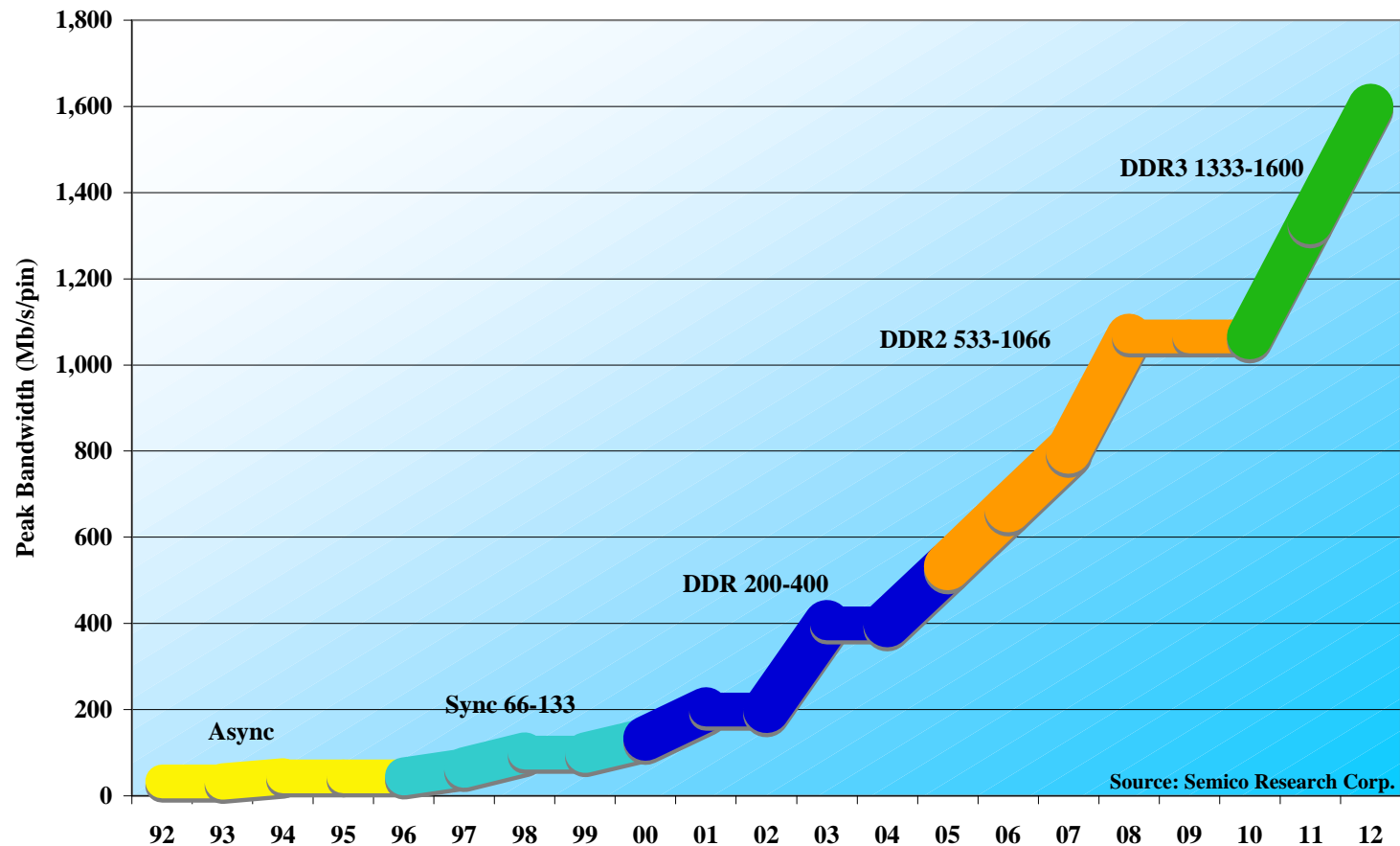
- More complex video formats: MPEG2 → H.264
- More complex video profiles: QCIF → CIF → 720p → 1080p
- Higher graphics performance for OSD and gaming

Agenda

- HQHD video SoCs
- ***High-bandwidth DRAMs***
- Delivering high DRAM bandwidth in video SoCs
- Experimental results

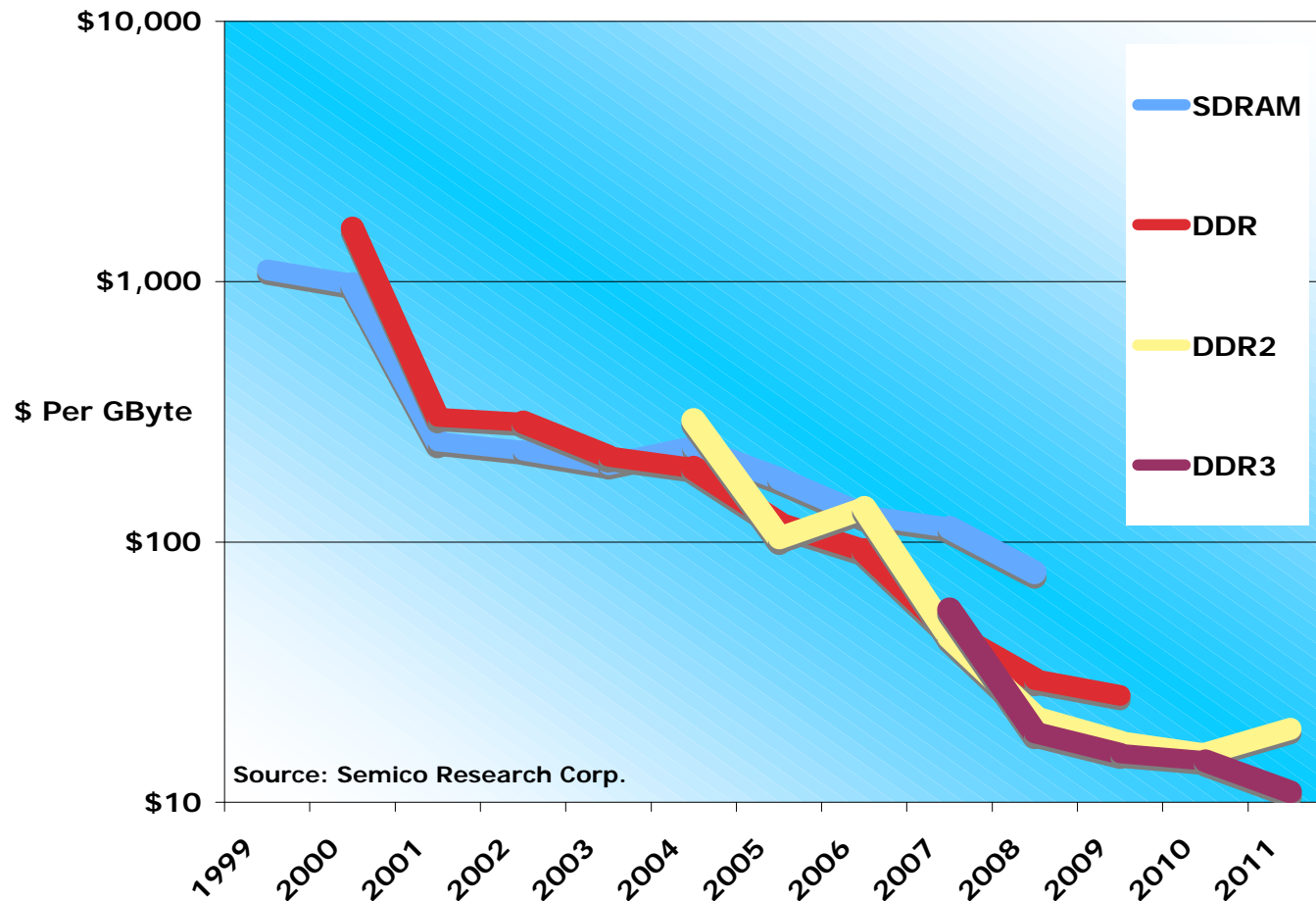
Why Video SoCs are Moving to DDR3

Peak Bandwidth Evolution



Why Video SoCs are Moving to DDR3

Average Cost Per GByte of DRAM Interface



High-Bandwidth DRAM Evolution

Appendix SDRAM Device Evolution

	SDR DRAM	DDR SDRAM	DDR2 SDRAM	Qimonda DDR3 SDRAM
Data rate [Mbit/s per pin]	PC66, PC100, PC133	DDR200, 266, 333, 400	DDR2-400, 533, 667, 800	DDR3-800, 1066, 1333, 1600
I/O organization	x4, x8, x16	x4, x8, x16	x4, x8, x16	x4, x8, x16
VDD = VDDQ [V]	3.3 (+/- 0.3)	2.5 (+/- 0.2)	1.8 (+/- 0.1)	1.5 (+/- 0.075)
Interface	LVTTL	SSTL_2	SSTL_18	SSTL_15
Number of Banks	2/4	4	4/8	8
Prefetch	1	2	4	8
Burst length	1, 2, 4, 8 (page)	2, 4, 8	4, 8	8 (step 4)
Bidirectional strobe	None	Single Ended (SE)	SE and Differential	Differential only
DQ driver strength	Wide envelope	Narrow envelope	18 Ω , ODD calibration	34 Ω , ZQ-pin cal
Termination		MoBo	MoBo/ODT	DDMM/Dynamic ODT
Read Latency	CL = (1), 2, 3	CL = (1.5), 2, 2.5, (3)	CL = (2), 3, 4, 5	CL = 5, 6, 7, 8, 9, 10, (11)
Additional Latency	-	-	AL = 0, 1, 2, 3, 4	AL = 0, CL-1, CL-2
Write Latency	0	1	RL-1	5, 6, 7, 8 + AL
Data mask	Yes	Yes	Yes	Yes
Interrupts	Yes	Yes	Wr-Wr, Rd-Rd 4n only	Bank only for Bank 0-7
Package	T50P-54	T50P-66/BGA	BGA	BGA

Increased
Bandwidth
but Increased
Minimum Bursts

Minimal Latency
Improvement

Source: Qimonda/Infineon

Optimizing High-Bandwidth DRAMs

■ Characteristics

- High peak bandwidth
- High latency
- Large minimum burst length
- High page miss cost
- High read/write data bus turnaround costs
- Multi-bank architecture supporting overlapped access

■ Optimizations

- Maximize page hits to minimize bank cycling costs
- Pipeline accesses to cover latency
- Exploit bank-level parallelism to hide page miss penalties
- Cluster reads and writes to minimize direction changes

■ Memory subsystem needs many outstanding transactions and out-of-order (O-O-O) service capabilities

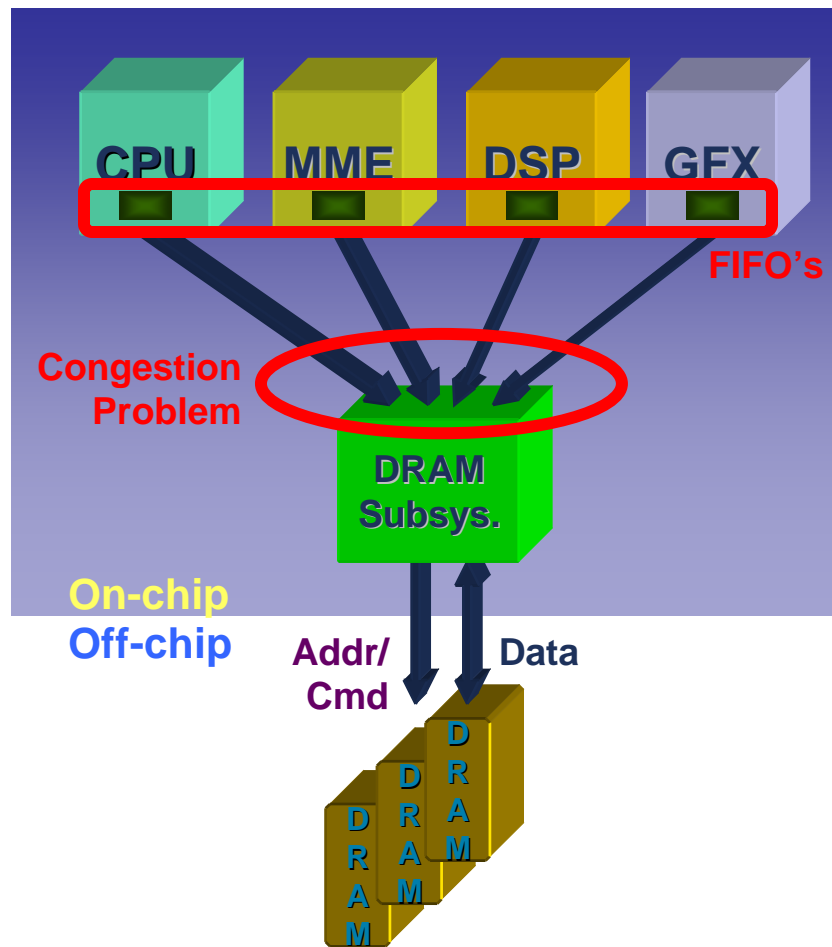
- Cortex A-series processors architected for such systems!

Agenda

- HQHD video SoCs
- High-bandwidth DRAMs
- ***Delivering high DRAM bandwidth in video SoCs***
 - ***Memory subsystem design***
 - ***System interconnect design***
- Experimental results

Star Topology Memory Subsystems

Traditional Approach



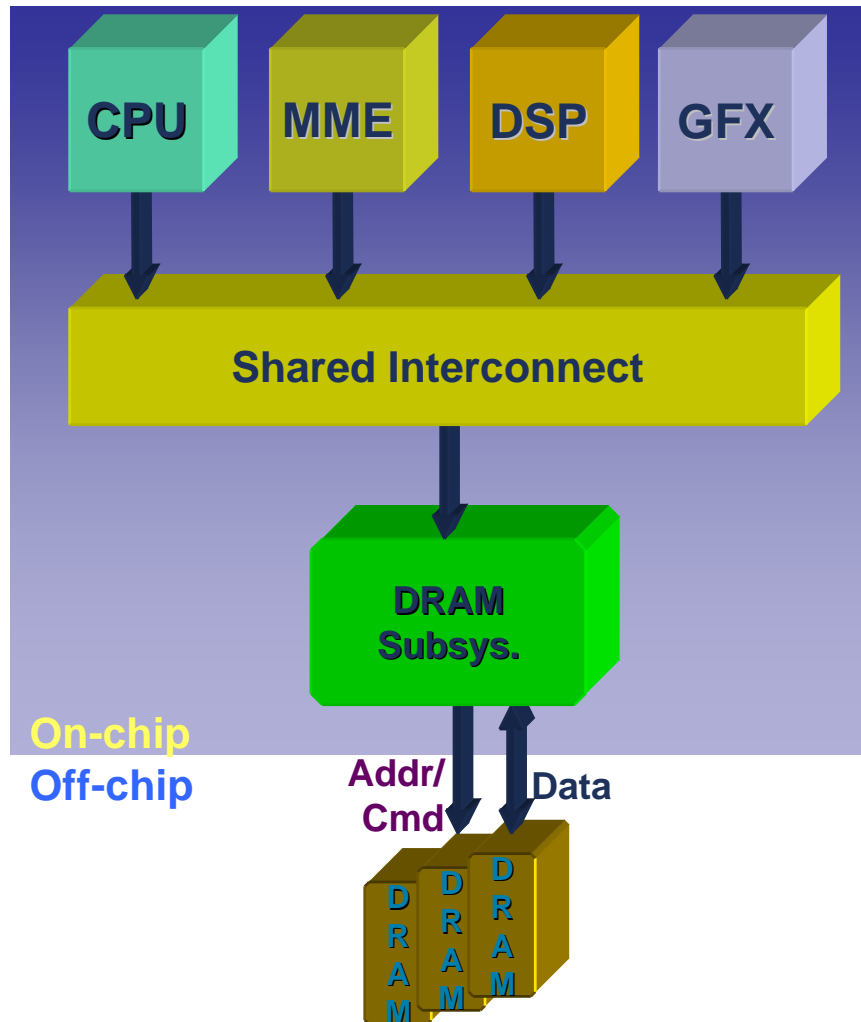
- Initiators present requests in parallel to multi-port scheduler
- FIFOs at initiators provide
 - Rate decoupling
 - Service jitter tolerance
- DRAM subsystem needs no FIFO, only pipelining
- System performance limited only by traffic and scheduler

But:

- LOTS of wires/congestion
- Lots of small/inefficient FIFOs
- Large part of system must be bandwidth matched to DRAM

Single-ported Memory Subsystems

Shared Interconnect Approach



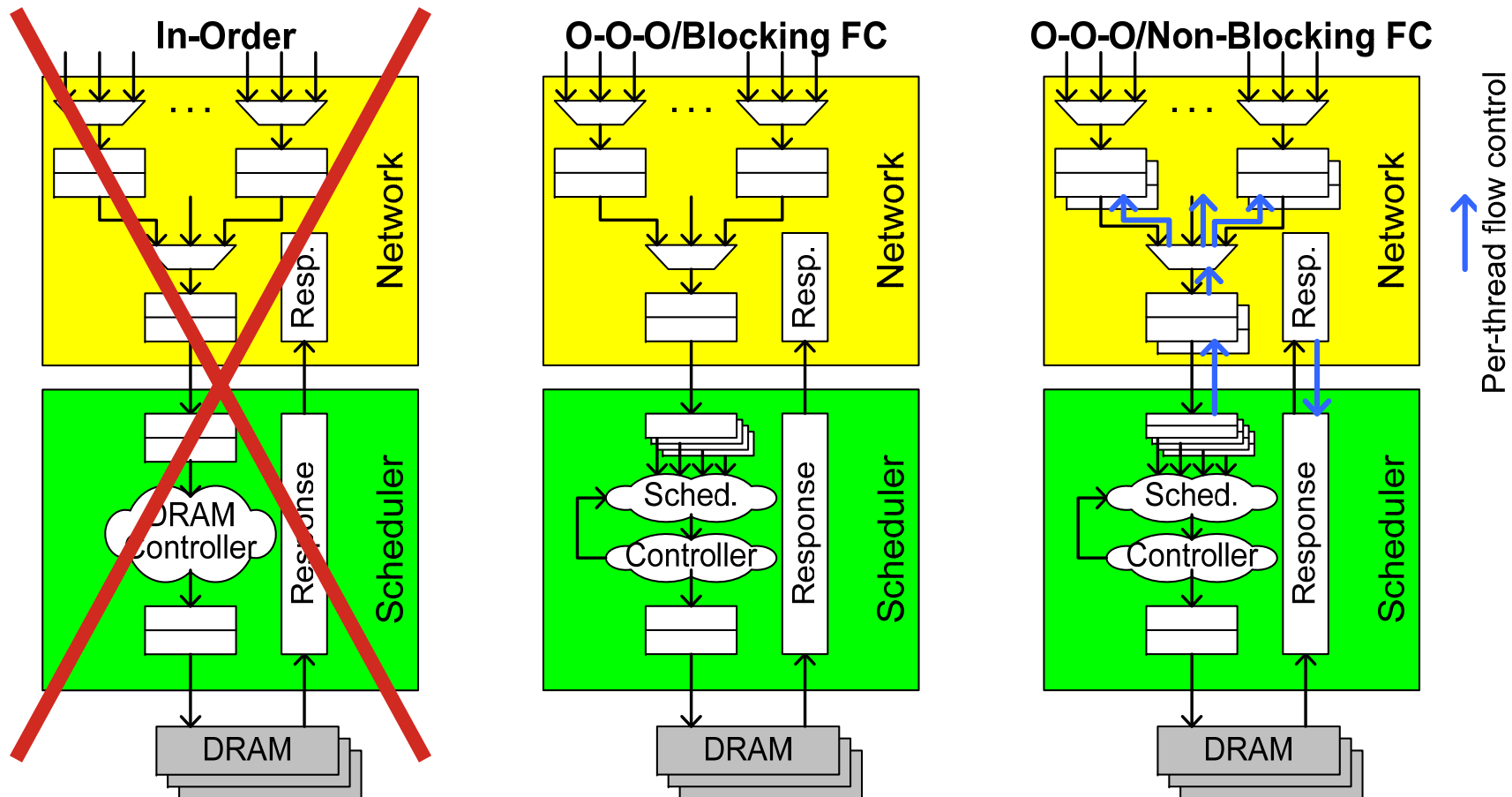
- Interconnect presents requests in series to single-port scheduler
- Saves wires/congestion

But:

- Interconnect arbitration impacts scheduling
 - Risks lower utilization
 - May not meet deadlines
- Where do FIFO's live?
- How much of system is bandwidth-matched to DRAM?
- System performance also limited by communication system

Single-port DRAM Protocols

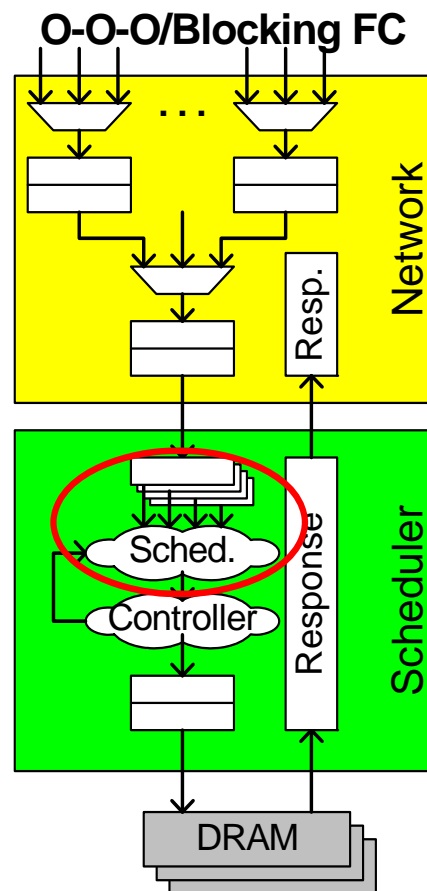
- Interconnect and subsystem must support multiple outstanding requests (cover DRAM pipeline depth)



Out-of-order Protocol with Blocking FC

- Interface protocol provides ordering tags to allow scheduler to reorder some requests, but flow control is shared across all tags

Examples:
AMBA AXI
OCP Tags



- Interconnect presents requests in order
- Scheduler queues requests and chooses order to optimize throughput and QoS

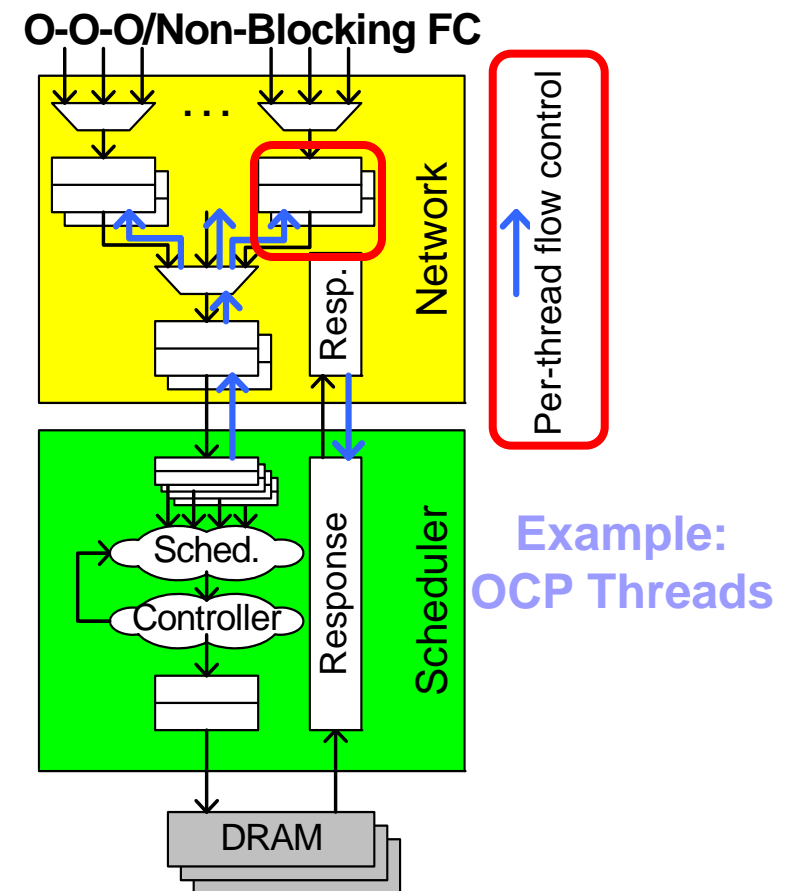
But

- Bursty flows can fill queues, hurting latency and bandwidth for others
- Full queues block into network
- Frequency scales poorly with queue depth

Out-of-order Protocol with Non-blocking FC

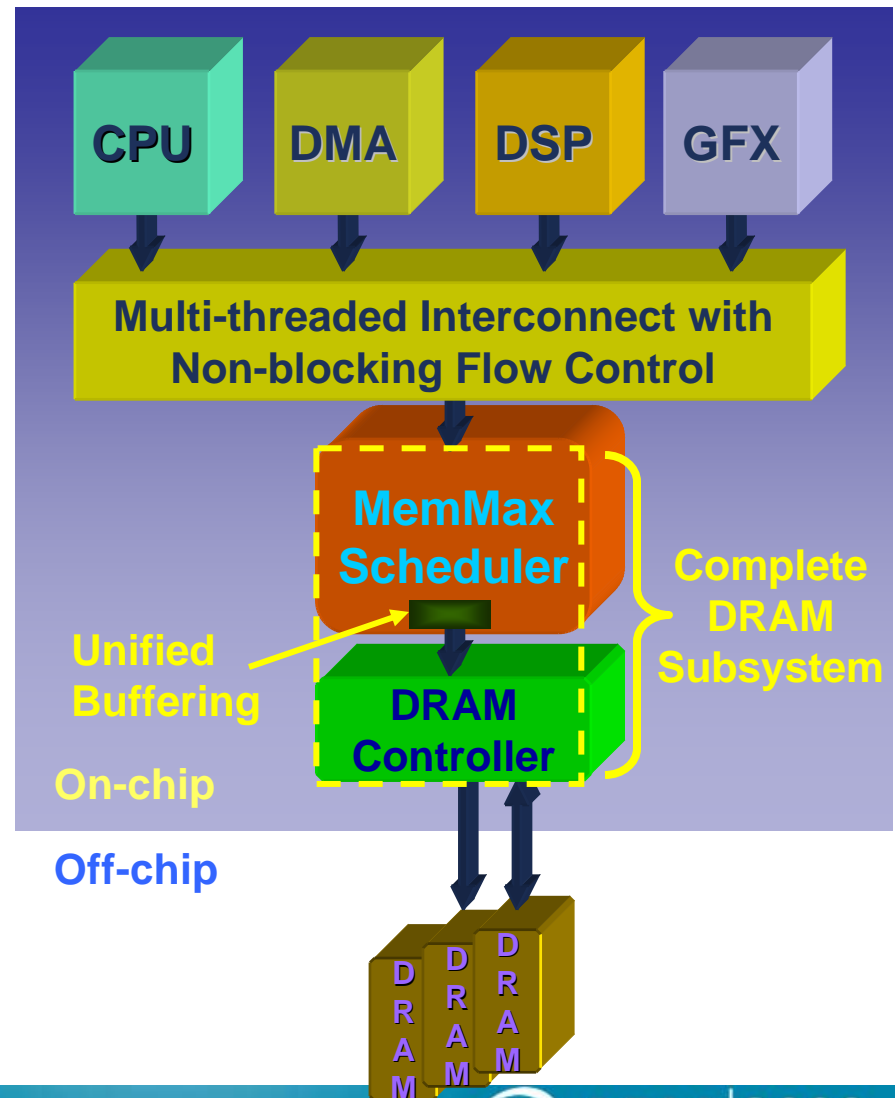
- Interface protocol provides per-thread ID's and flow control, enabling re-ordering while preventing blocking

- Interconnect maps initiator threads into target threads
- Scheduler queues requests and chooses order to optimize throughput and QoS *on per-thread basis*
- Non-blocking (per-thread) flow control minimizes inter-thread interactions
- Per-thread queues inherently ordered, implemented as compiled SRAM
- Result: lower latency, bandwidth guarantees, higher guaranteed throughput

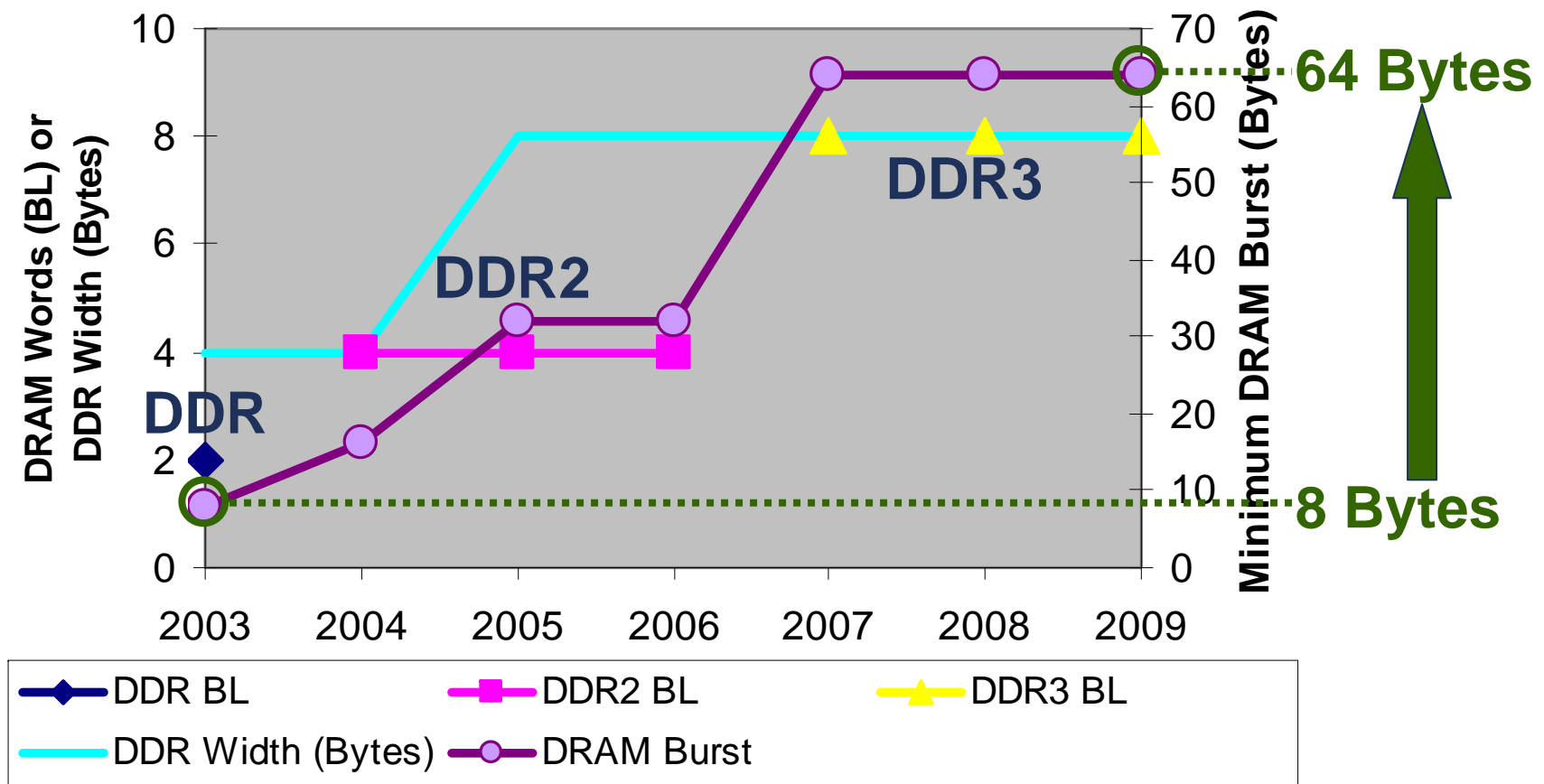


Example Memory Subsystem Solution

- MemMax Memory Scheduler
 - Single port scheduler
 - Multi-threaded interface protocol support out-of-order with non-blocking flow control
 - Unified SRAM-based buffering provides performance decoupling
 - Optimizes DRAM efficiency while guaranteeing QoS
- High performance system interconnect
- Generic DRAM controller

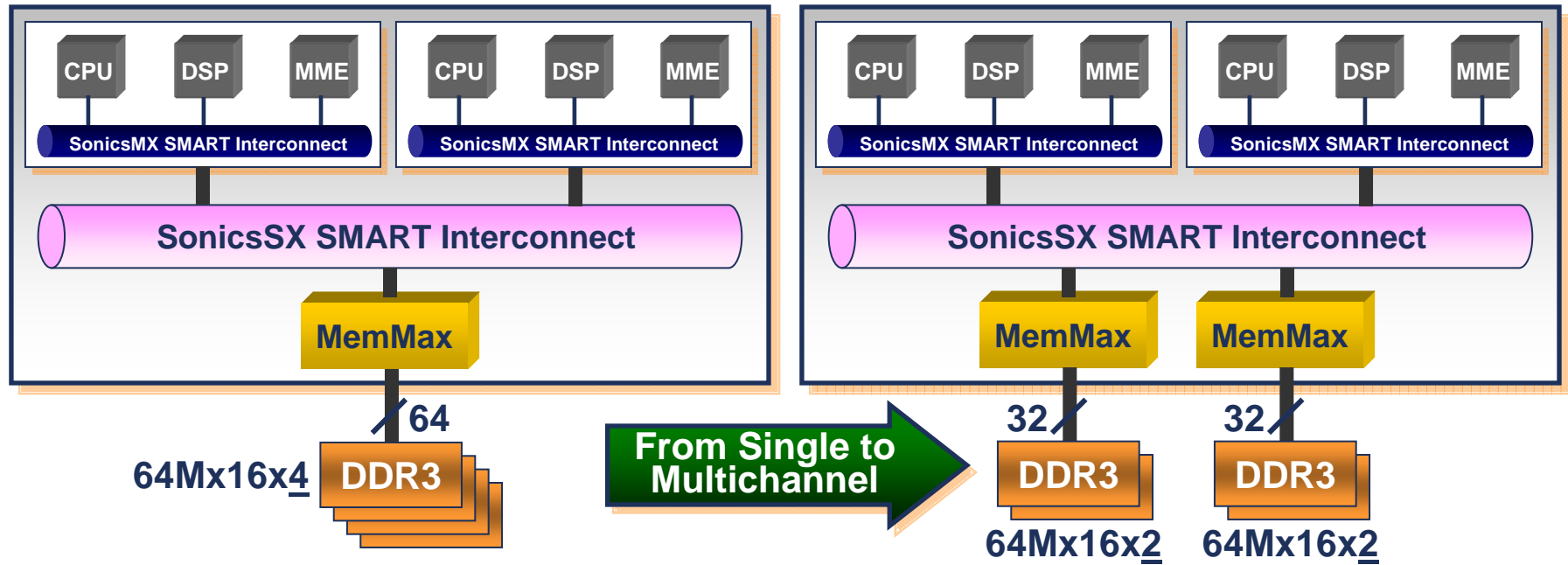


DRAM Burst Sizes Growing Too Large



Many SoC Data Objects \leq 32 Bytes!

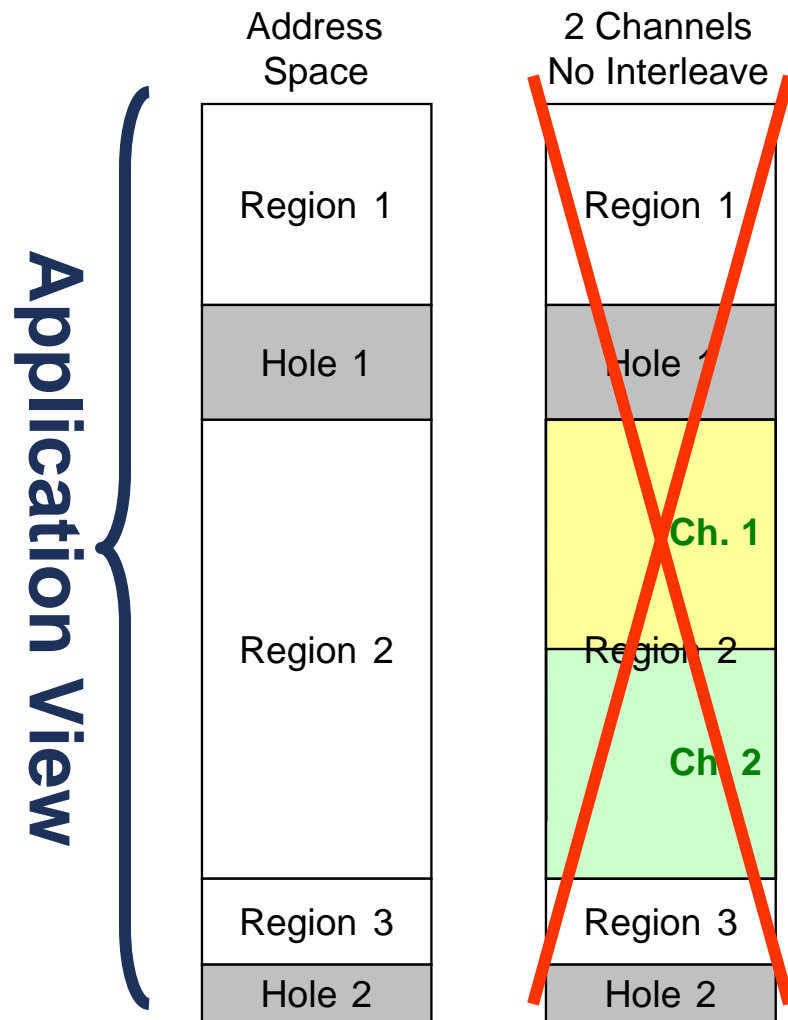
Multichannel Optimizes DRAM Efficiency



	DDR2	DDR3	DDR3
Channels	1	1	2
Data Width (B)	4	4	2
Effective BW	100%	84%	100%

Source: Customer (HDTV) System Dataflow
Constant Frequency/Ideal Load Balancing

Multichannel Is Not Easy!

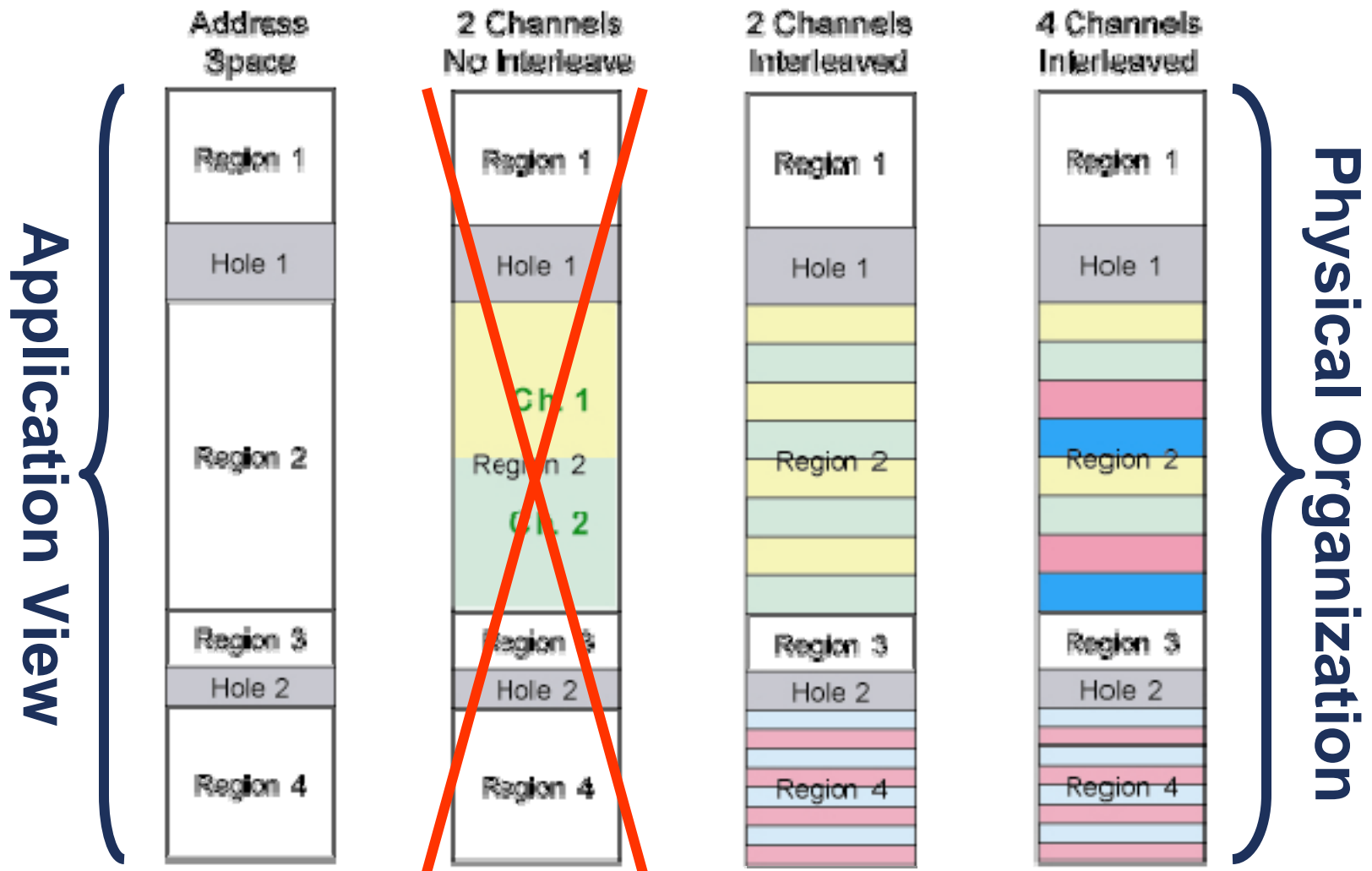


Key Problems:

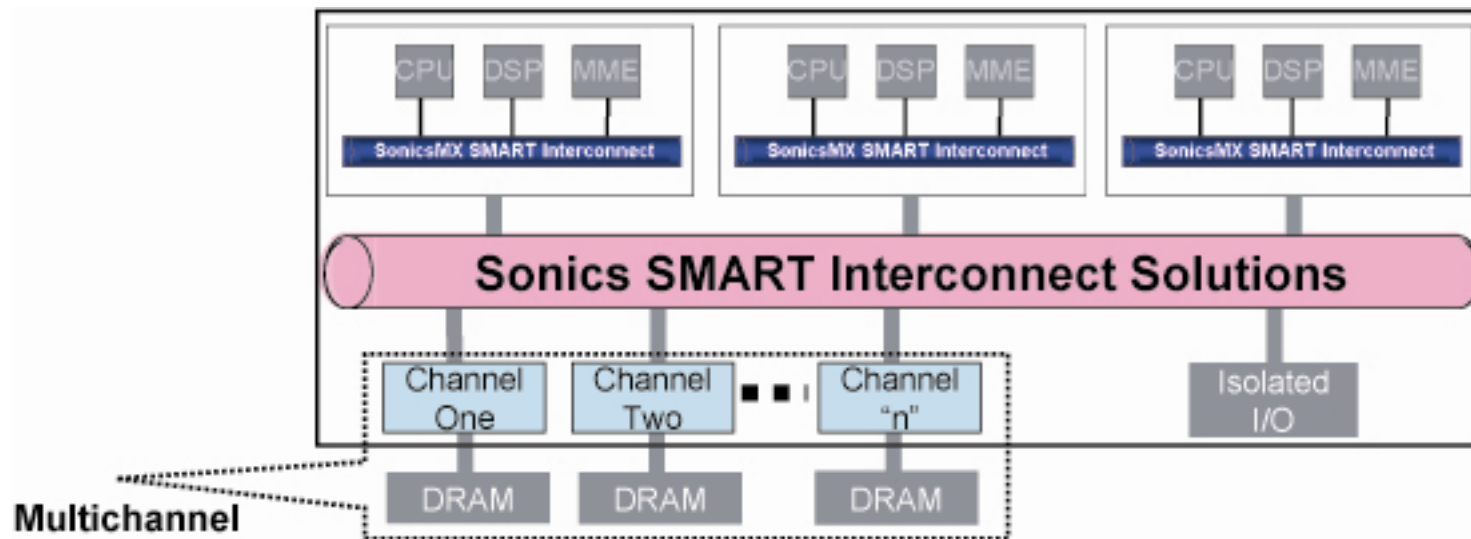
- Load balancing
 - Must balance memory traffic evenly among channels
- Maintaining throughput
 - Multiple channels cause throughput/ordering problems for pipelined memories

This means software and IP cores must manage multiple channels

Seamless Multichannel Transition



SonicsSX SMART Interconnect Solution



- High performance 16 GB/s interconnect
 - Supports 2D transfers
 - Supports external address tiling
- Includes new IMT™ (interleaved multichannel technology)
 - Manages load balancing and channel splitting automatically

Multichannel Interleaving in the Interconnect

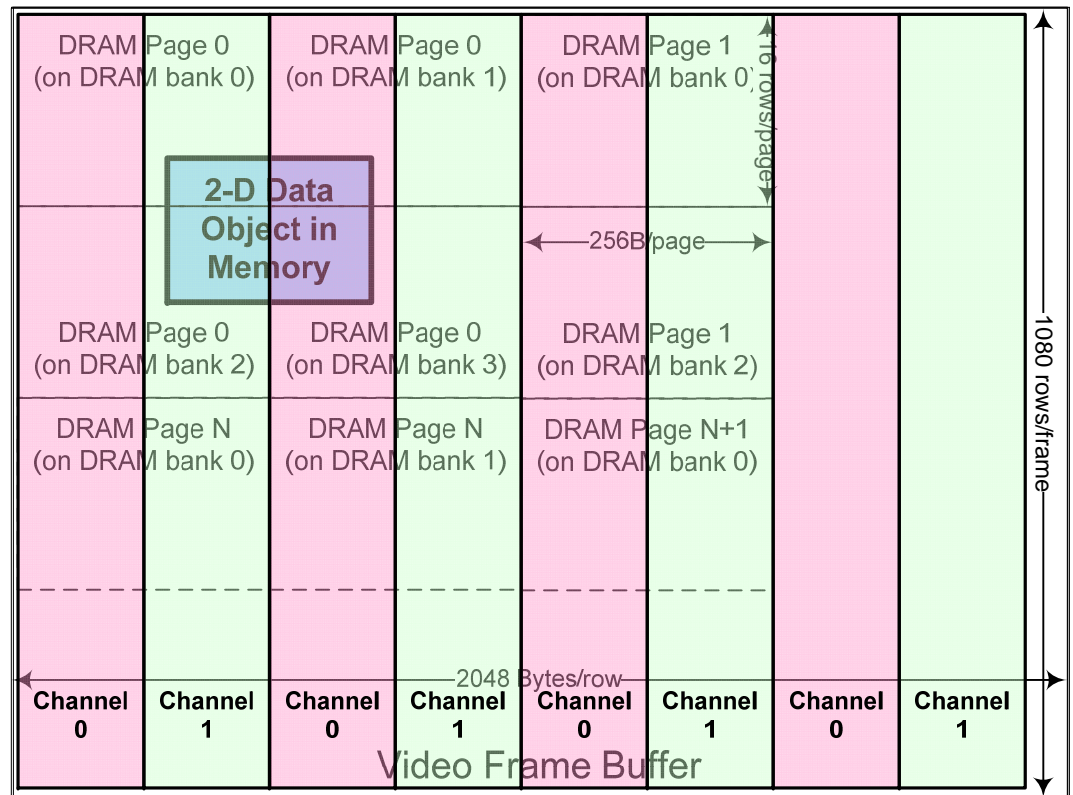
Higher Performance, Lower Area, More Scalable

- Interleaving support requires splitting traffic for delivery to proper channel
- Splitting in memory scheduler/controller
 - Creates performance bottleneck
 - Hard to scale past two channels
- IMT splitting*
 - Fully distributed architecture enables scalability
 - Network overlaps channel accesses to maximize throughput
 - Optimized protocols eliminate reorder buffer area
 - Isolating channels from IP cores makes it transparent to software and other hardware

*patents pending

2D Bursts, Address Tiling & Multichannel

- Two-dimensional block bursts
 - 2D transaction using a single read/write command
 - Popular for HD video and graphics
- Address tiling
 - Rearrange DRAM address organization to exploit 2D locality
 - Avoids page misses
- Channels divide buffer into columns
 - SonicsSX splits 2D bursts that cross channel edges



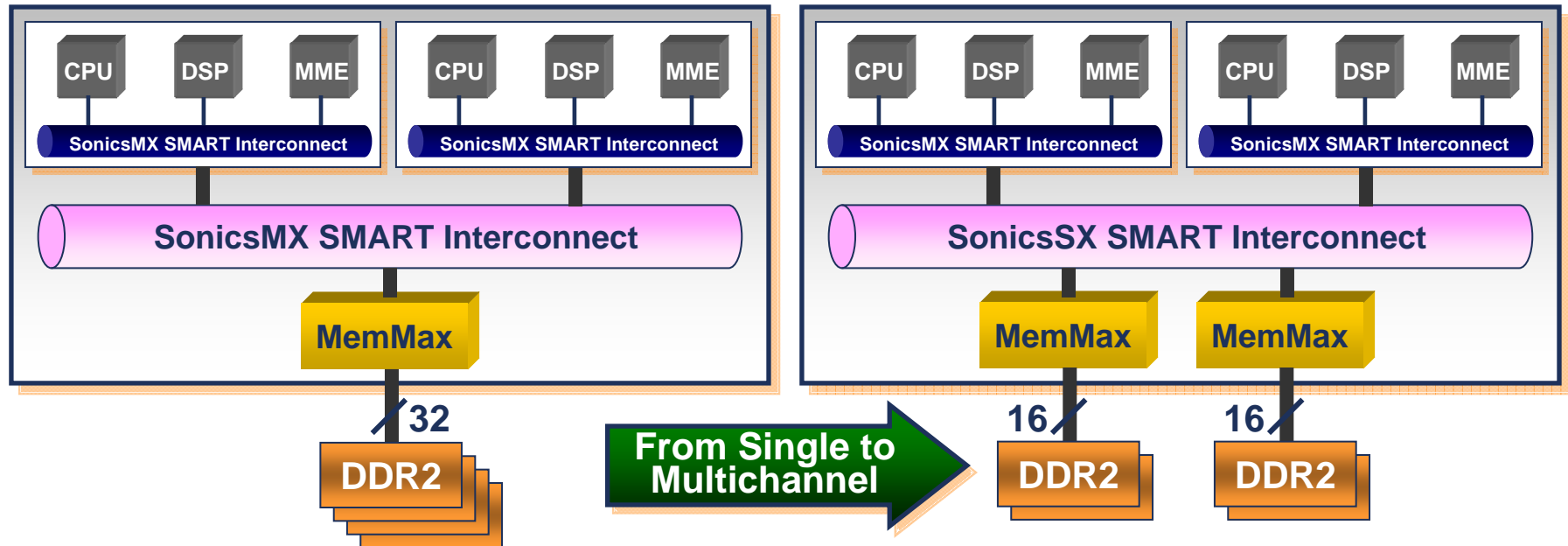
SonicsSX Meets Video SoC Memory Needs

- Advanced architecture provides predictability
 - Leverages multithreading and guaranteed QoS of SMART Interconnect solutions
- IMT provides scalable memory performance
 - Supports up to eight channels of DRAM
 - Automatic load balancing and channel management
- Transparent to hardware and software
 - Decouples IP cores and software from the memory configuration

Agenda

- HQHD video SoCs
- High-bandwidth DRAMs
- Delivering high DRAM bandwidth in video SoCs
- ***Experimental results***

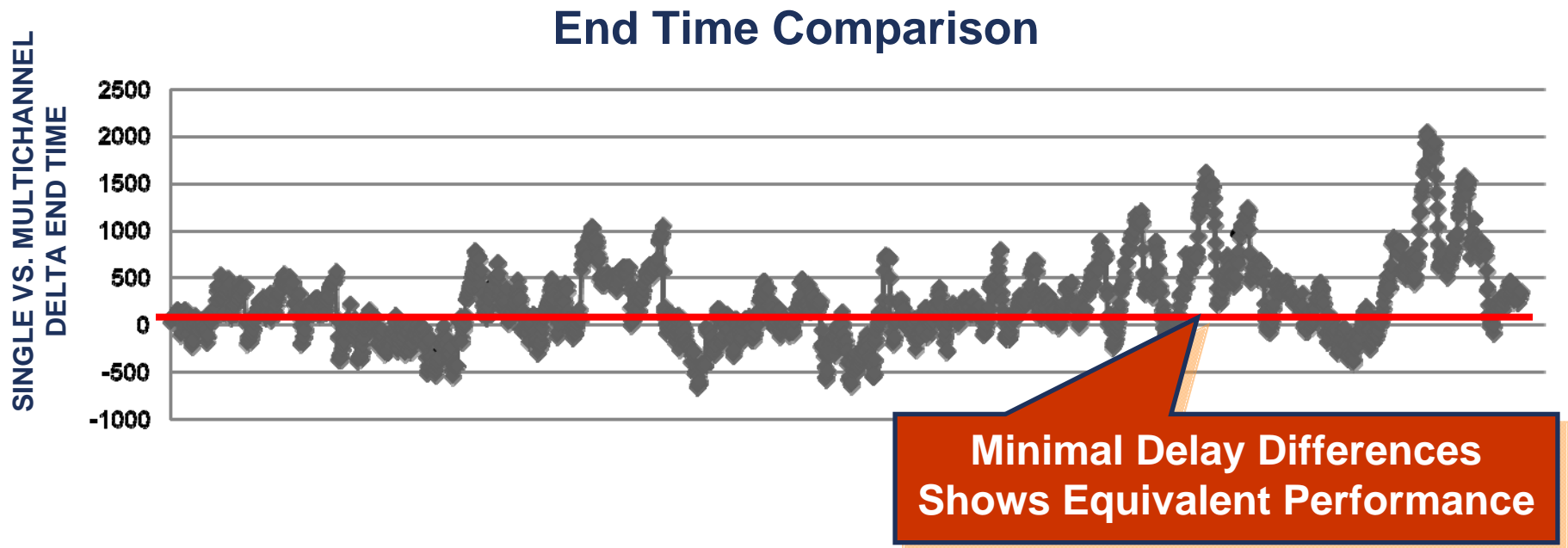
Test Case: Production HDTV SoC



- Objective: compare existing production single channel design to multichannel design ***with the same total bandwidth***
 - Prove IMT makes multichannel performance transparent to application
- Analysis based on 30 ms of customer-provided trace data
- Measure differences between transaction end times

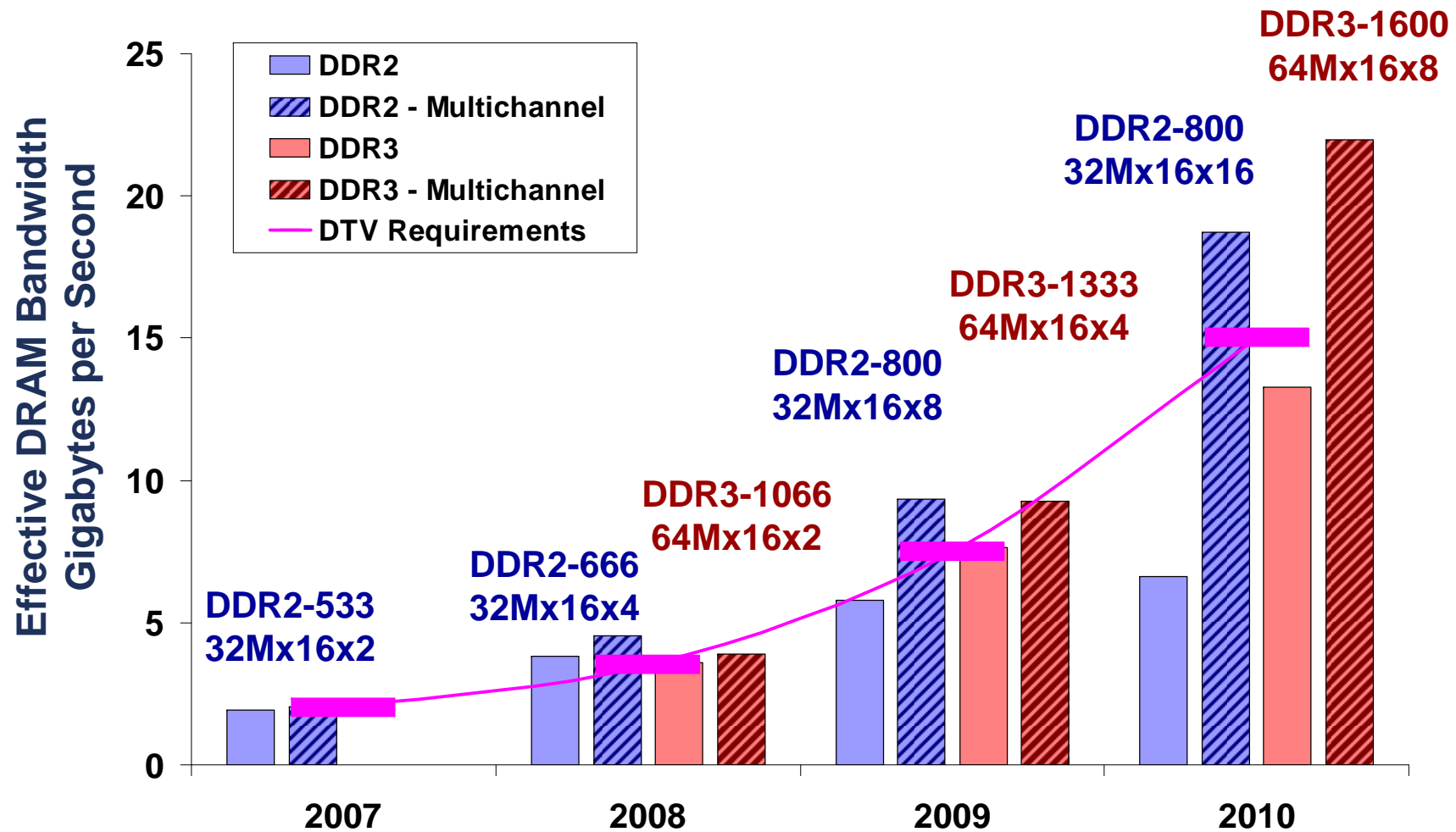
IMT Run Time Analysis

(Example: Worst case segment for video frame)



IMT: Transparent Multichannel Performance

Multichannel Meets HQHD TV Requirements



Summary

- Video SoCs are characterized by:
 - Lots of processors communicating with each other via DRAM
 - High DRAM throughput requirements at high efficiency
 - These problems get tougher each generation
- Video SoC architecture dominated by system interconnect and memory subsystems
 - Deeply pipelined, out-of-order DRAM scheduling is needed that protects QoS while optimizing efficiency
 - Multichannel DRAM systems required to maintain efficiency
- Key technologies improve performance and predictability while reducing costs of HQHD video SoCs
 - Multi-threading with non-blocking flow control
 - Multichannel interleaving in the interconnect