Texas Instruments

TMS320C6678 (Shannon) DSP Training

Brighton Feng

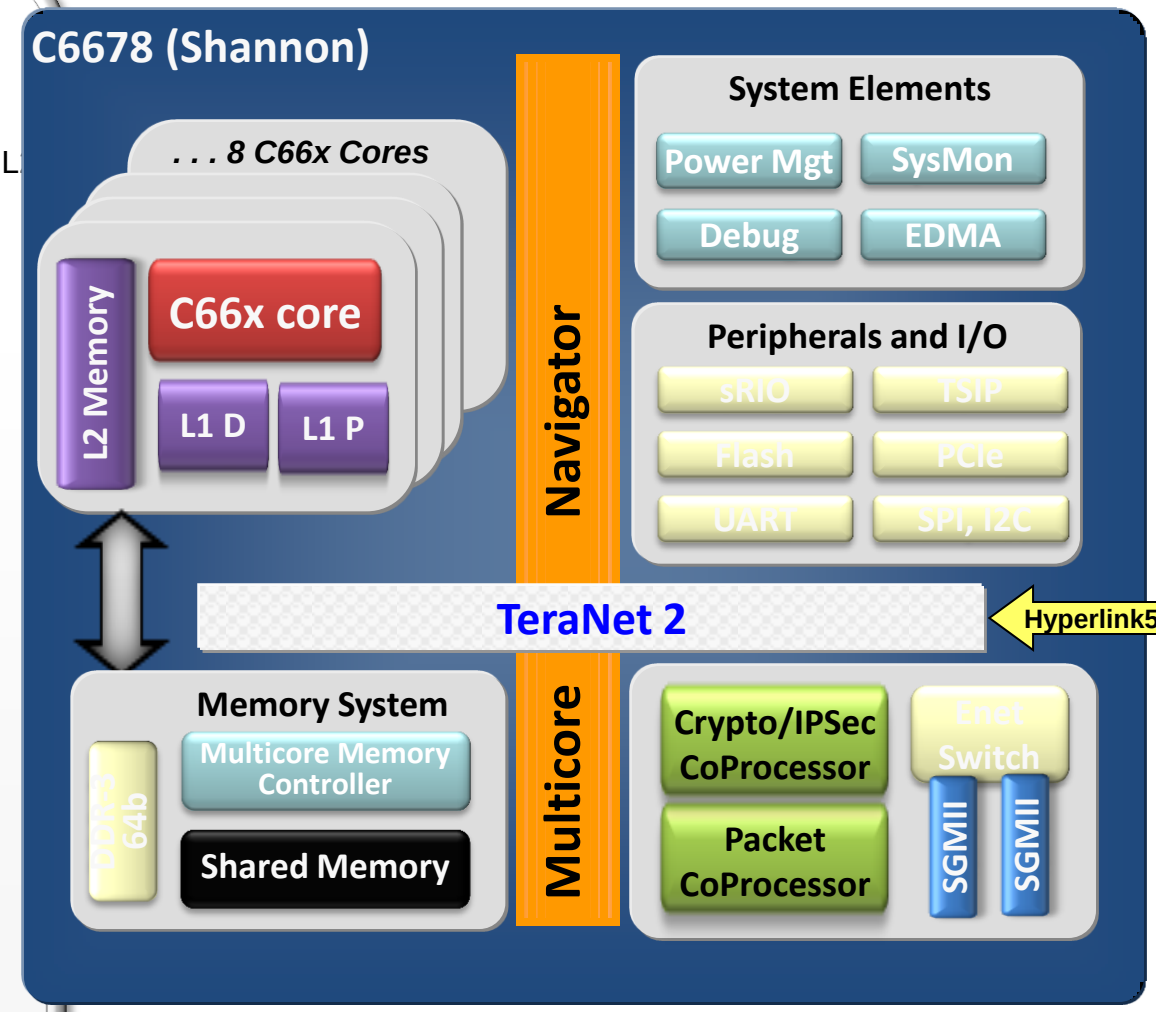
November, 2010

Outline

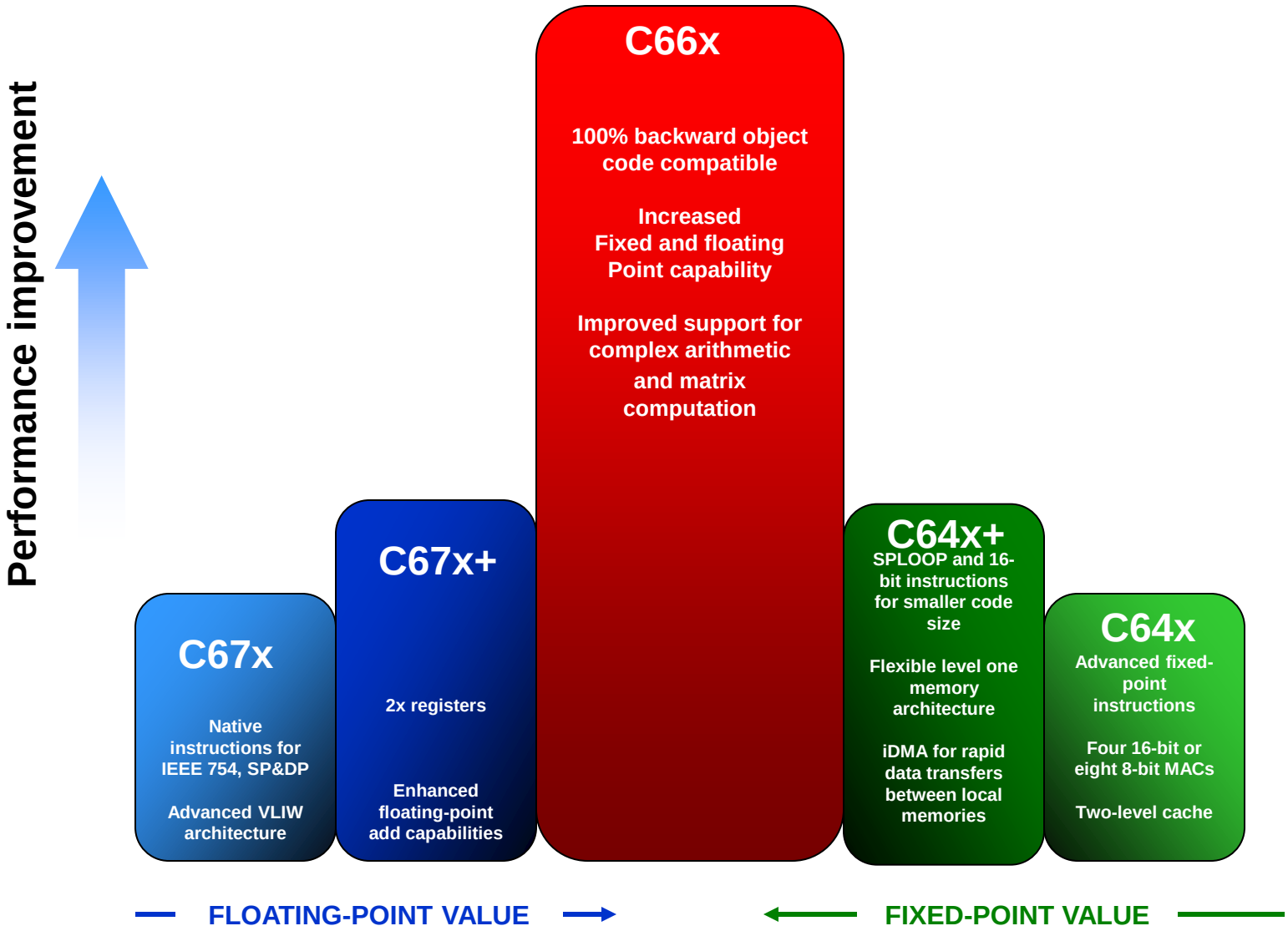
- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
- Peripherals overview

Shannon Functional Diagram

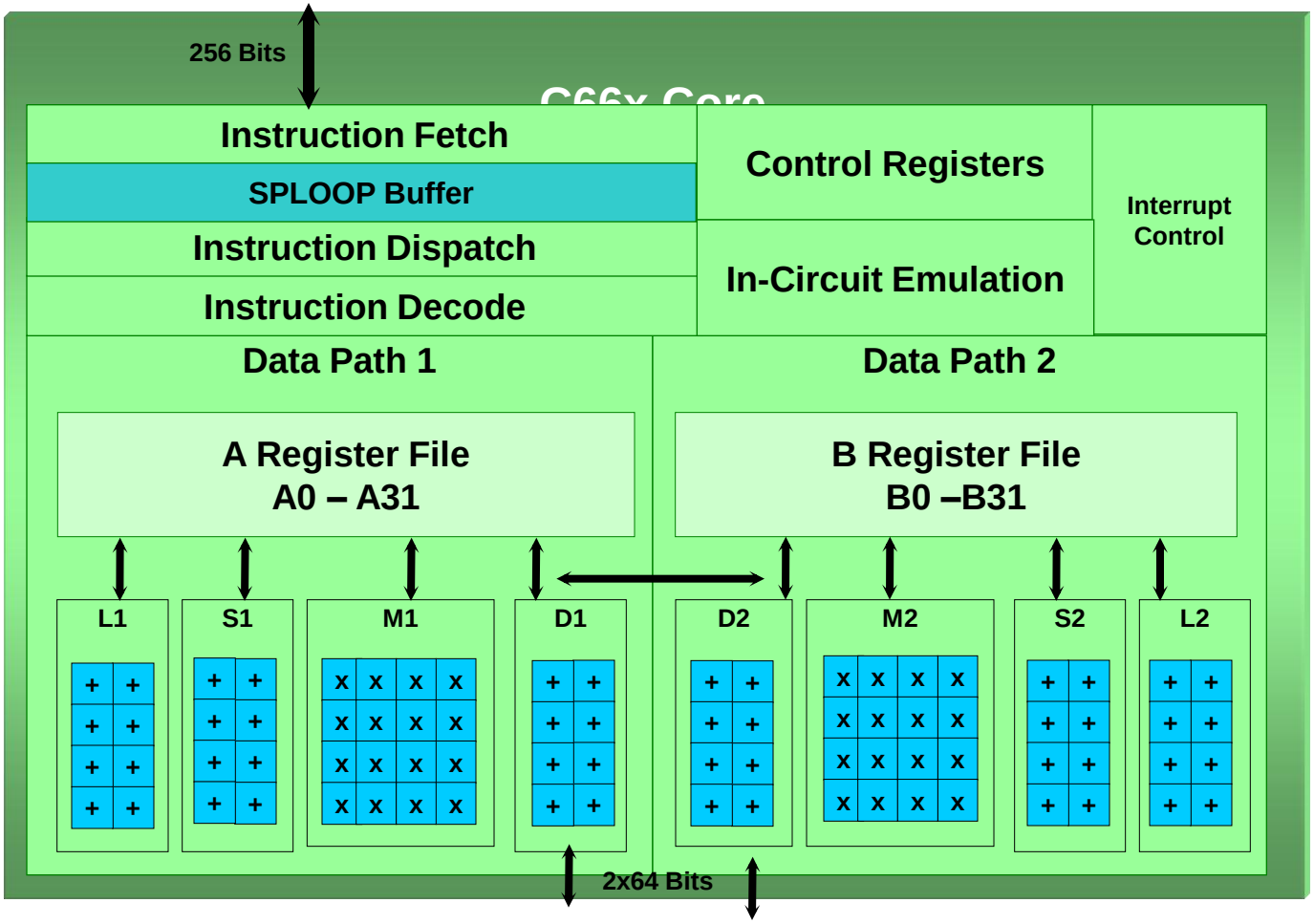
- Multi-Core SoC
- Fixed/Floating C66x™ Core
  - Eight cores @ 1.0 GHz, 0.5 MB Local L
  - 4.0 MB shared memory
  - 256 GMAC, 128 GFLOP
- Navigator
  - Multicore eco system
    - Packet Infrastructure
- Network Coprocessor
  - IP Network solution for IP v4/6
  - 1.5M packets per sec (1Gb Ethernet wire-rate)
  - IPsec, SRTP, Air Interface Encryption fully offloaded
- 3-port GigE Switch (Layer 2)
- Low Power Consumption
  - Adaptive Voltage Scaling (Smart Reflex™)
- Hyperlink 50
  - 50G Expansion port
  - Transparent to Software
- Multicore Debugging



Enhanced DSP core



C66x core block diagram



Key Improvements of C66x

- ✓ 4x Multiply Accumulate improvement
  - ✓ Enhanced complex arithmetic and matrix operations
- ✓ 2x Arithmetic and Logical operations improvement
- ✓ Support the floating point arithmetic. Single precision floating point operation capability same as 32 bit fixed point operation capability
- ✓ division and square root is supported by floating point instruction

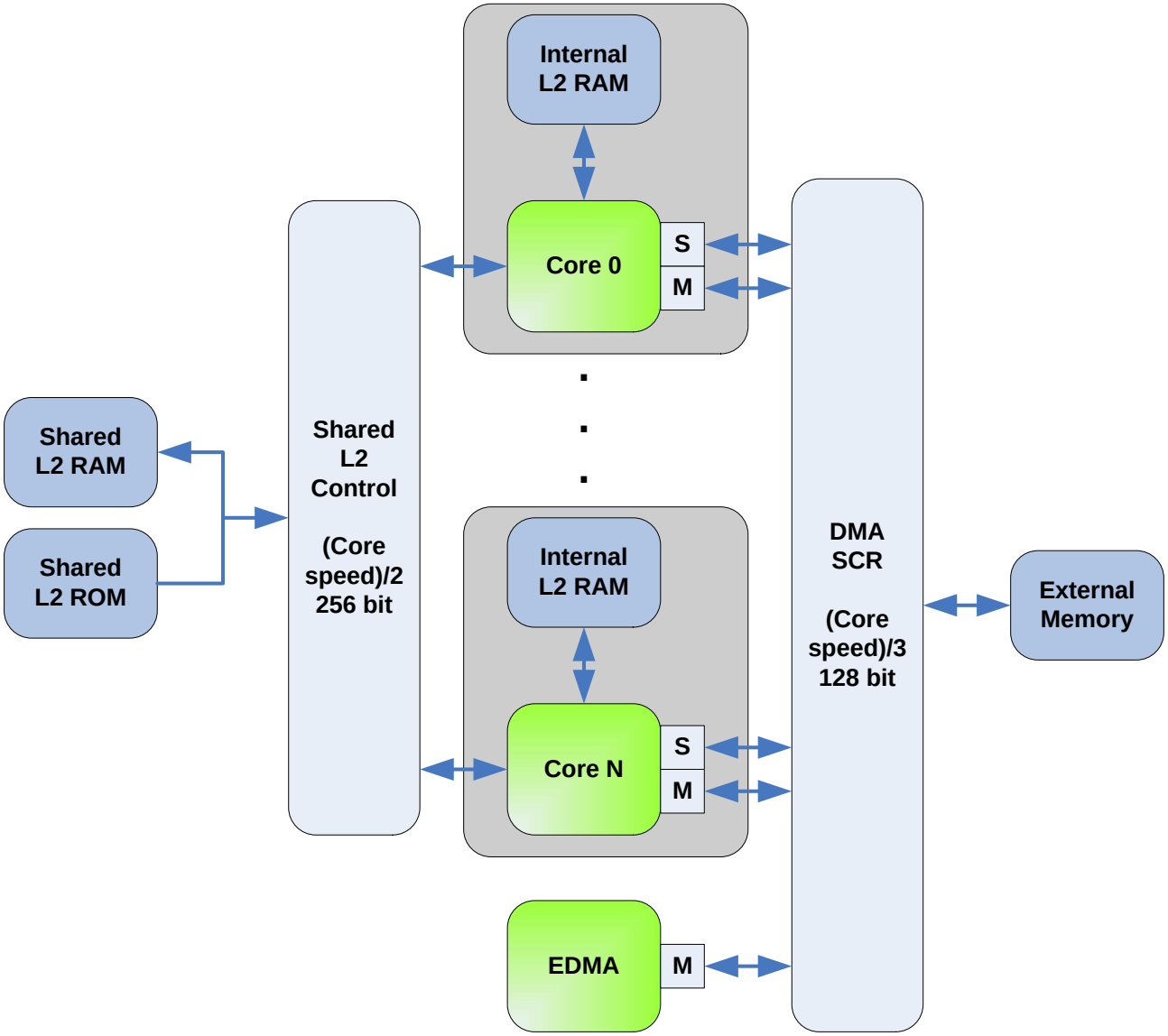
C64x+ → C66x Comparison

Operation	Precision	Operations per cycle on C64x+	Operations per cycle on C66x	Function Unit
MAC	Real 8 x 8	2 x 4 = 8	2 x 8 = 16	M1, M2
	Real 16 x 16	2 x 2 = 4	2 x 8 = 16	M1, M2
	Real 32 x 32	2 x 1 = 2	2 x 4 = 8	M1, M2
	Complex (16,16) x (16,16)	2 x 1 = 2	2 x 4 = 8	M1, M2
	Complex (32,32) x (32,32)	N/A	2 x 1 = 2	M1, M2
Arithmetic Logical	8 bit	4 x 4 = 16	4 x 8 = 32	L1, L2, S1, S2
	16 bit	4 x 2 = 8	4 x 4 = 16	L1, L2, S1, S2
	32 bit	4 x 1 = 4	4 x 2 = 8	L1, L2, S1, S2
Memory Access	8 bit, 16 bit, 32 bit, 64 bit	2 x 1 = 2	2 x 1 = 2	D1, D2

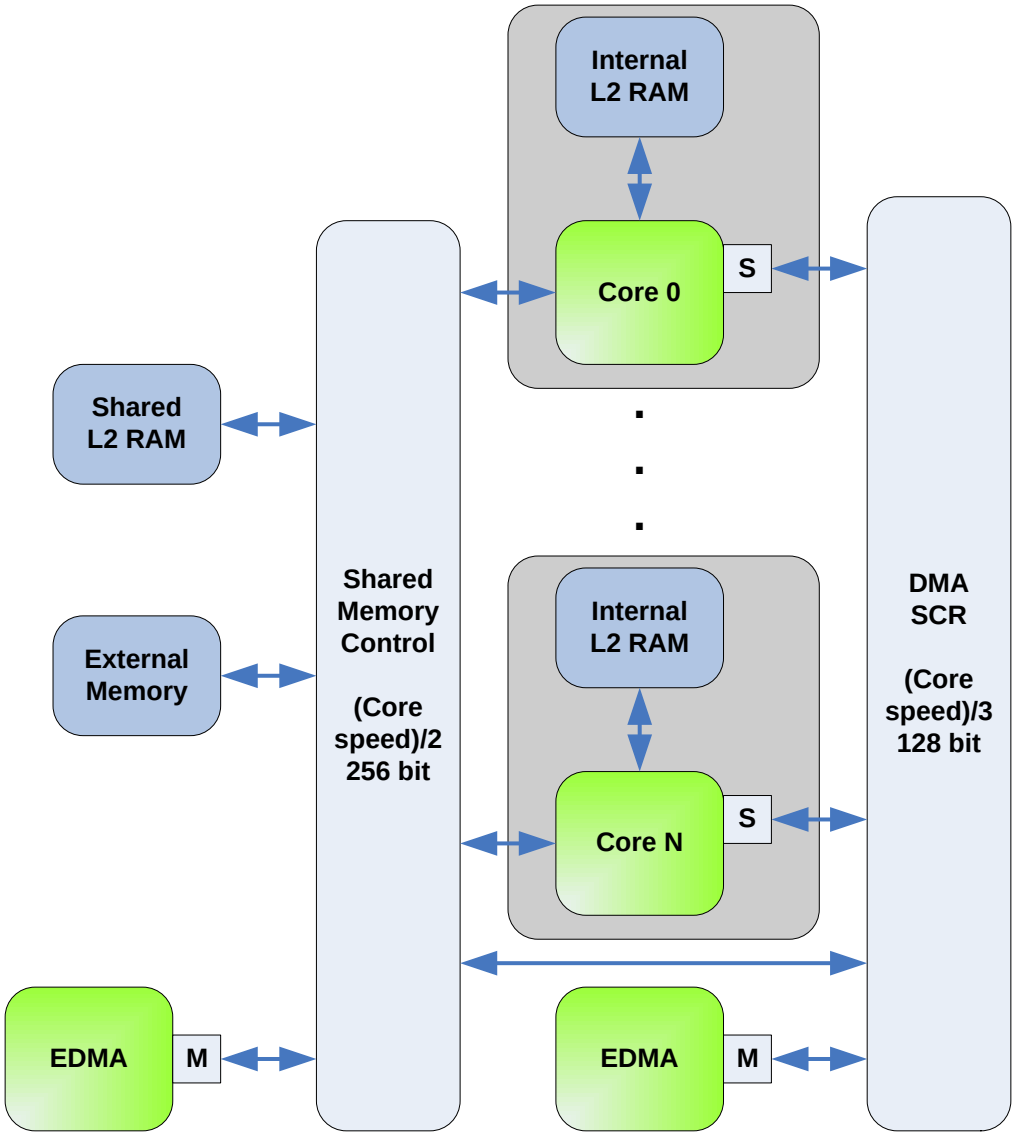
Outline

- C6678 DSP Overview
- Multi-core DSP programming
  - Memory Architecture Overview
  - Shannon Memory Architecture Improvement
  - Programming model
- Interconnection and resource sharing
- Peripherals overview

TCI6486 Memory Architecture



Shannon Memory Architecture



Outline

- C6678 DSP Overview
- Multi-core DSP programming
  - Memory Architecture Overview
  - Shannon Memory Architecture Improvement
  - Programming model
- Interconnection and resource sharing
- Peripherals overview

Addition of XMC

- Bring over existing EMC MDMA path
- Fat pipe to external (and internal) shared memory
  - Bus width: 256 instead of 128 bits
  - Clock rate: CPUCLK/2 instead of CPUCLK/3
  - Optimize requests for MSMC / DDR3 memory
    - L2 line allocations and evictions are split into sub-lines of 64 bytes
- Memory Protection and Address Extension (MPAX) support
  - 16 segments of programmable size (powers of 2: 4KB to 4GB)
  - Each segment maps a 32-bit address to a 36-bit address.
  - Each segment controls access: supervisor/user, R/W/X, (non-)secure
  - Memory protection for shared internal MSMC memory and external DDR3 memory
- Multi-stream Prefetch support
  - Program prefetch buffer up to 128 bytes
  - Data prefetch buffer up to 8 x 128 bytes
  - Prefetch enabled/disabled on 16MB ranges defined in MAR
  - Manual flush for coherence purposes
- Note: no IDMA path

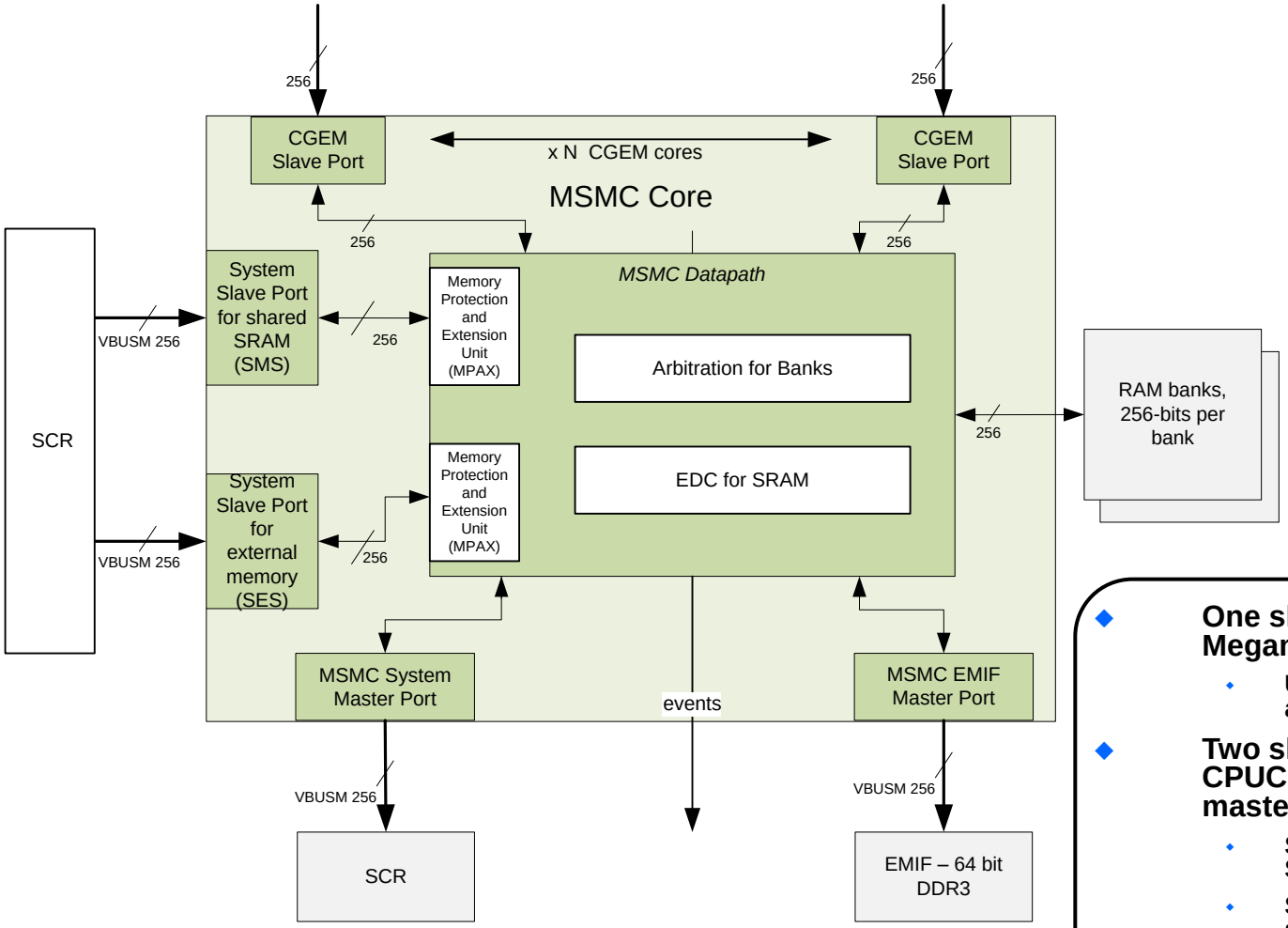
MAR Register Extension

• L2 memory controller extends the MAR registers by adding the “PFX” field, L2 memory controller uses this bit to convey XMC whether a given address range is prefetchable.

Figure 1-2 MAR Register Layout

	31				4	3	2	1	0	
MARn	Reserved					PFX	Rsvd	Rsvd	PC	0184_8*xxx*
	R, +0000 0000 0000 0000 0000 0000 0000					RW	R, +0	R, +0	RW	

MSMC Block Diagram



- One slave interface per C66x Megamodule (256 bits @ CPUCLK/2)
  - Uses a 36 bit address extended inside a C66x Megamodule core
- Two slave interfaces (256 bits @ CPUCLK/2) for access from system masters
  - SMS interface for accesses to MSMC SRAM space
  - SES interface for accesses to DDR3 space
  - Both interfaces support memory protection and address extension
- One master interface (256-bits @ CPUCLK/2) for access to the DDR3 EMIF
- One master interface (256 bits @ CPUCLK/2) for access to system slaves

MSMC Shared Memory

- 4 banks x 2 sub-banks, sub-bank are 256-bit wide.
  - Reduces conflicts between C66x Megamodule cores and system masters
  - Features a dynamic fair-share bank arbitration for each transfer
- Supports bandwidth management. Avoid indefinite starvation for lower priority requests due to higher priority requests
- Features Not Supported
  - Cache coherency between L1/L2 caches in C66x Megamodule cores and MSMC memory
  - Cache coherency between XMC prefetch buffers and MSMC memory
  - C66x Megamodule to C66x Megamodule cache coherency via MSMC

MPAX Units

- MPAX stands for “Memory Protection and Address Extension”
- There are N+2 MPAX units in a system with N C66x Megamodules
  - N MPAX units for all requests from N C66x Megamodules to internal shared memory, external shared memory or any system slave
  - 1 MPAX unit for all requests from any system master to internal shared memory
  - 1 MPAX unit for all requests from any system master to external shared memory
- Each MPAX unit operates on a number of segments of programmable size
  - Each segment maps a 32-bit address to a 36-bit address.
  - Each segment controls access.

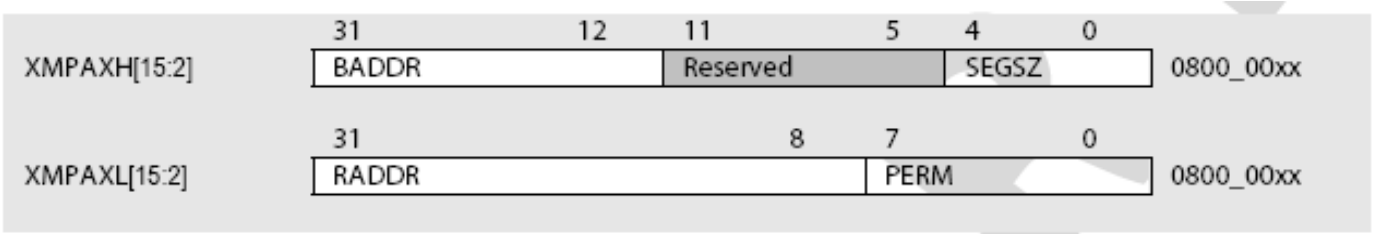
Number of Segments

- Each C66x Megamodule has 16 segments which control direct (load/store) requests to internal shared memory, external shared memory and any other system slave.
- Any master identified by a privilege ID has
  - 8 segments for requests to internal shared memory
  - 8 segments for requests to external shared memory.
- Some masters work on behalf of other masters. They will inherit the privilege ID of their commanding master. As such, each C66x Megamodule also has
  - 8 segments for indirect (DMA) requests to internal shared memory
  - 8 segments for indirect (DMA) requests to external shared memory

Segment Definition

- Each segment is defined by a base address and a size
  - The segment size can be set to any power of 2 from 4K to 4GB
  - The segment base address is constrained to power-of-2 boundary equal to size.
- One would expect that each request should find one matching segment, however ...
  - a request may find two or more matching segments, in which case segments with higher ID take priority over segments with lower ID. This allows
    - creating non-power of 2 segments
    - creating 3 segments with just 2 segment definitions
    - ...
  - a request may find no matching segment, in which case an error is reported in Memory protection fault reporting registers (XMPFAR, XMPFSR)

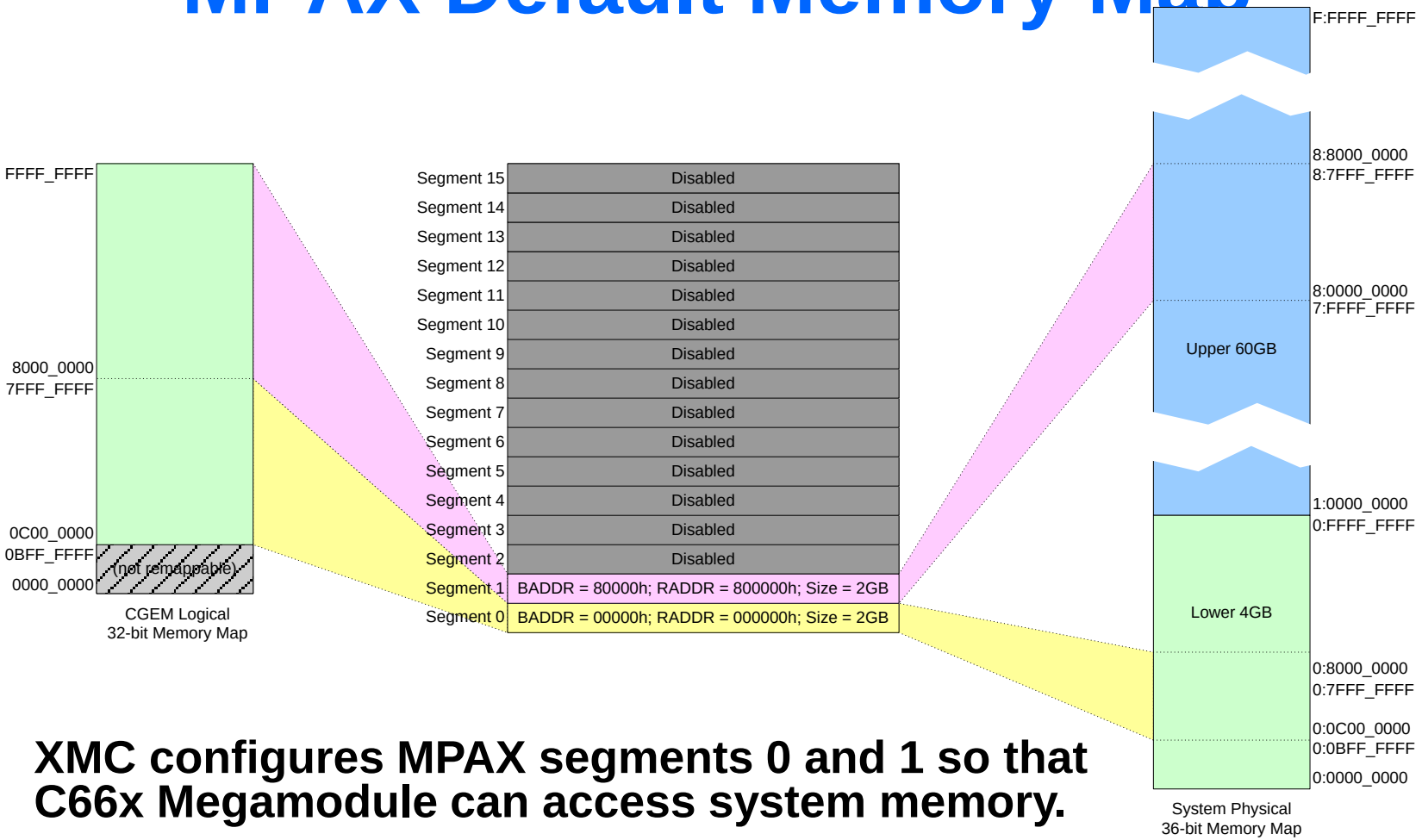
XMC Segment Registers XMPAXH/XMPAXL[15-0]



Field	Name	Meaning
BADDR	Base Address	Upper bits of address range to match in CGEM's native 32-bit address space
SEGSZ	Segment Size	Segment size. Table below indicates encoding.
RADDR	Replacement Address	Bits that replace and extend the upper address bits matched by BADDR
PERM	Permissions	Access types allowed in this address range.

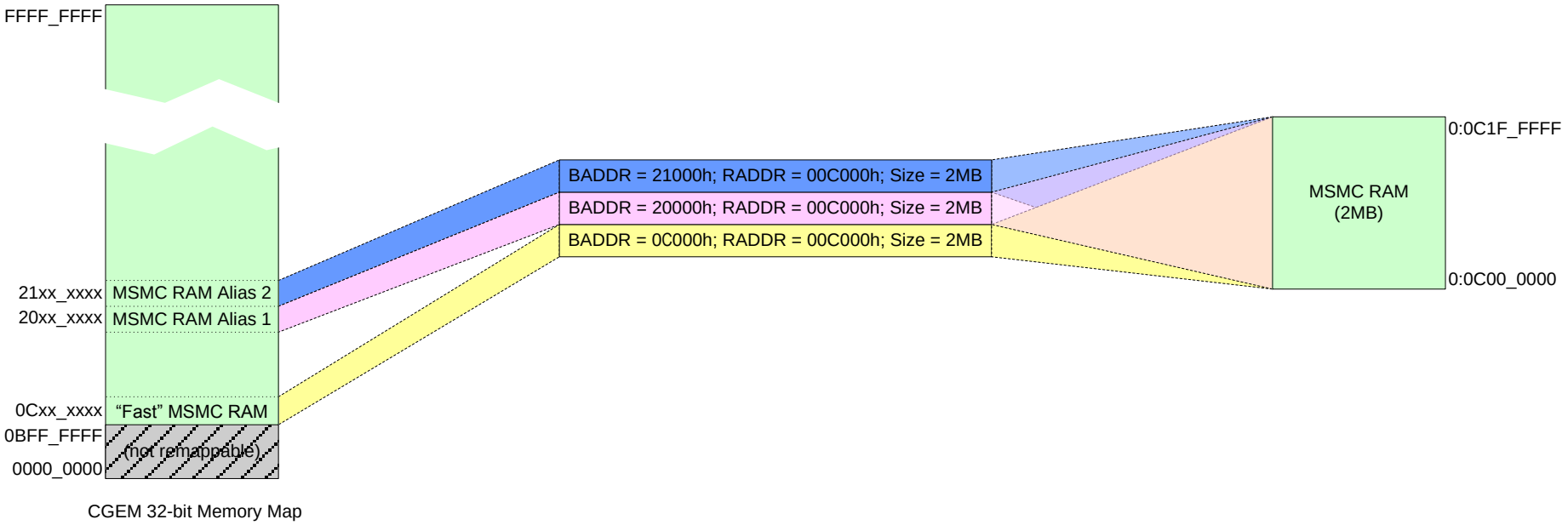
SEGSZ	Meaning	SEGSZ	Meaning	SEGSZ	Meaning	SEGSZ	Meaning
00000b	Seg. disabled	01000b	Rsvd (Disabled)	10000b	128KB	11000b	32MB
00001b	Rsvd (Disabled)	01001b	Rsvd (Disabled)	10001b	256KB	11001b	64MB
00010b	Rsvd (Disabled)	01010b	Rsvd (Disabled)	10010b	512KB	11010b	128MB
00011b	Rsvd (Disabled)	01011b	4KB	10011b	1MB	11011b	256MB
00100b	Rsvd (Disabled)	01100b	8KB	10100b	2MB	11100b	512MB
00101b	Rsvd (Disabled)	01101b	16KB	10101b	4MB	11101b	1GB
00110b	Rsvd (Disabled)	01110b	32KB	10110b	8MB	11110b	2GB
00111b	Rsvd (Disabled)	01111b	64KB	10111b	16MB	11111b	4GB

MPAX Default Memory Map



- **XMC configures MPAX segments 0 and 1 so that C66x Megamodule can access system memory.**
- **The power up configuration is that segment 1 remaps 8000_0000 – FFFF_FFFF in C66x Megamodule's address space to 8:0000_0000 – 8:7FFF_FFFF in the system address map.**
  - **This corresponds to the first 2GB of address space dedicated to EMIF by the MSMC controller.**

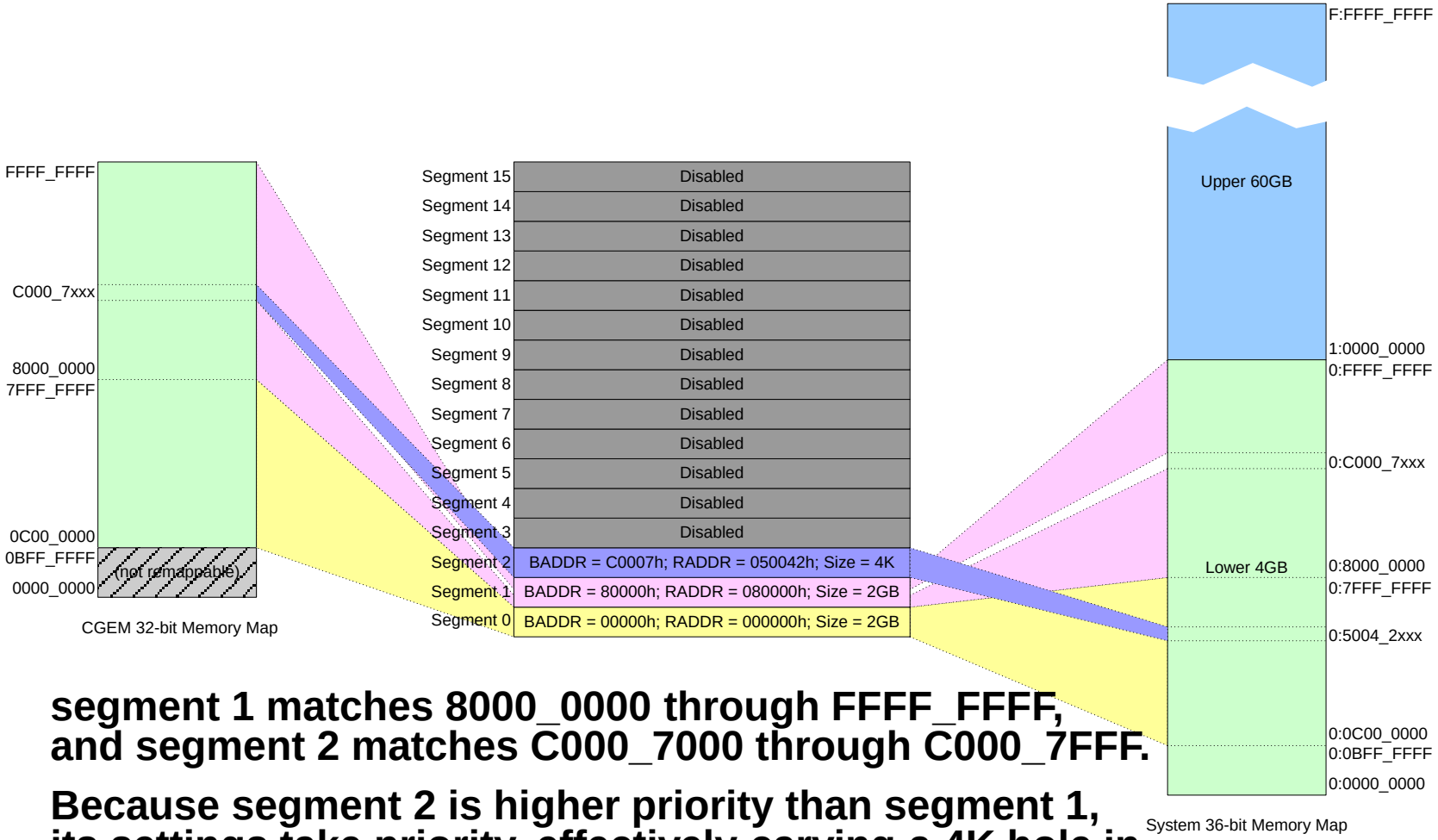
MPAX MSMC Aliasing Example



CGEM 32-bit Memory Map

- **Example shows 3 segments to map the MSMC RAM address space into C66x Megamodule's address space as three distinct 2MB ranges. By programming the MARs accordingly, the three segments could have different semantics.**
- **Accesses to MSMC RAM via this alias do not use the "fast RAM" path and incur additional cycles of latency.**

MPAX Overlayed Segments Example

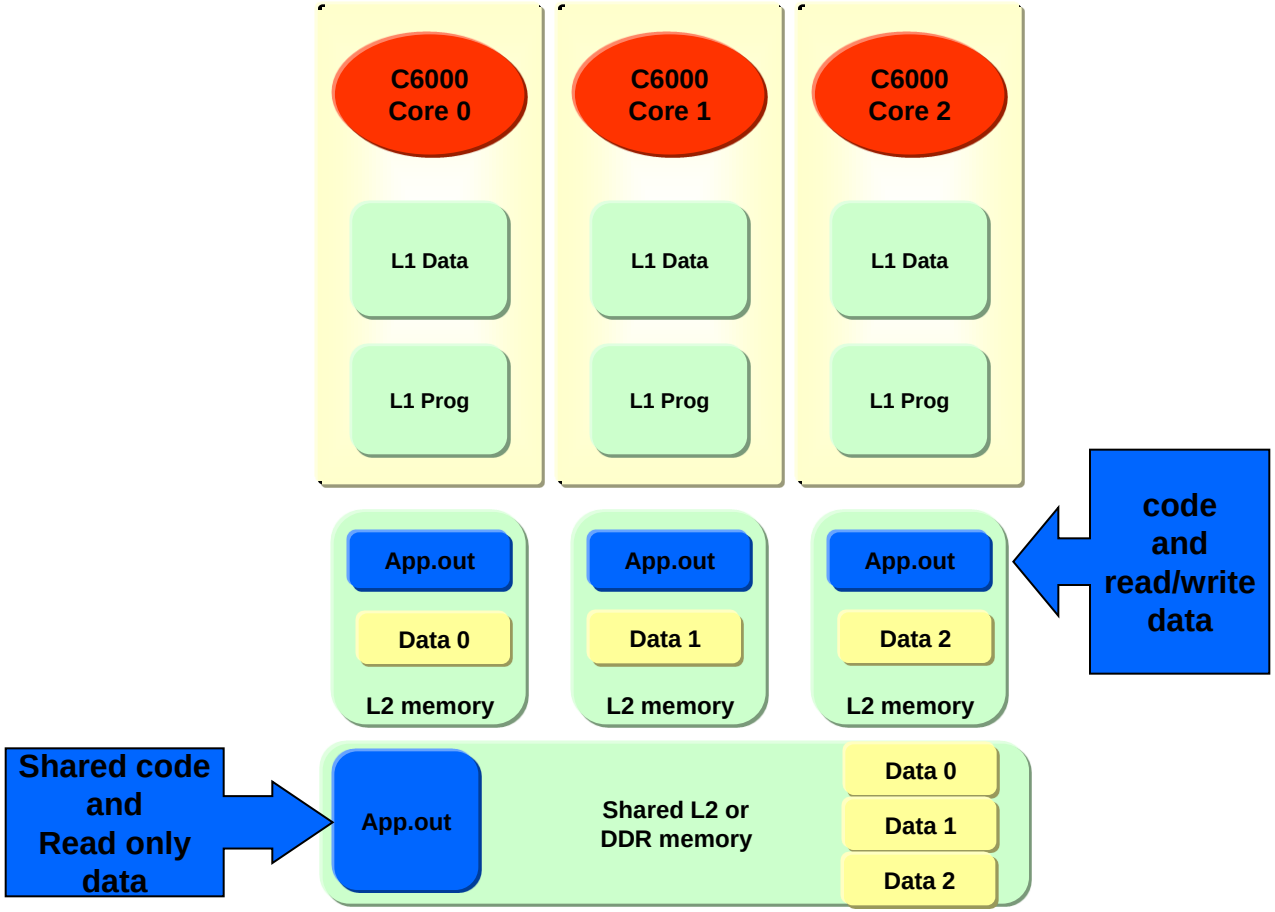


- **segment 1 matches 8000_0000 through FFFF_FFFF, and segment 2 matches C000_7000 through C000_7FFF.**
- **Because segment 2 is higher priority than segment 1, its settings take priority, effectively carving a 4K hole in segment 1's 2GB address space.**
- **Furthermore, it maps this 4K space to 0:5004_2000 - 0:5004_2FFF, which overlaps the mapping established by segment 0. This physical address range is now accessible by two logical address ranges.**

Outline

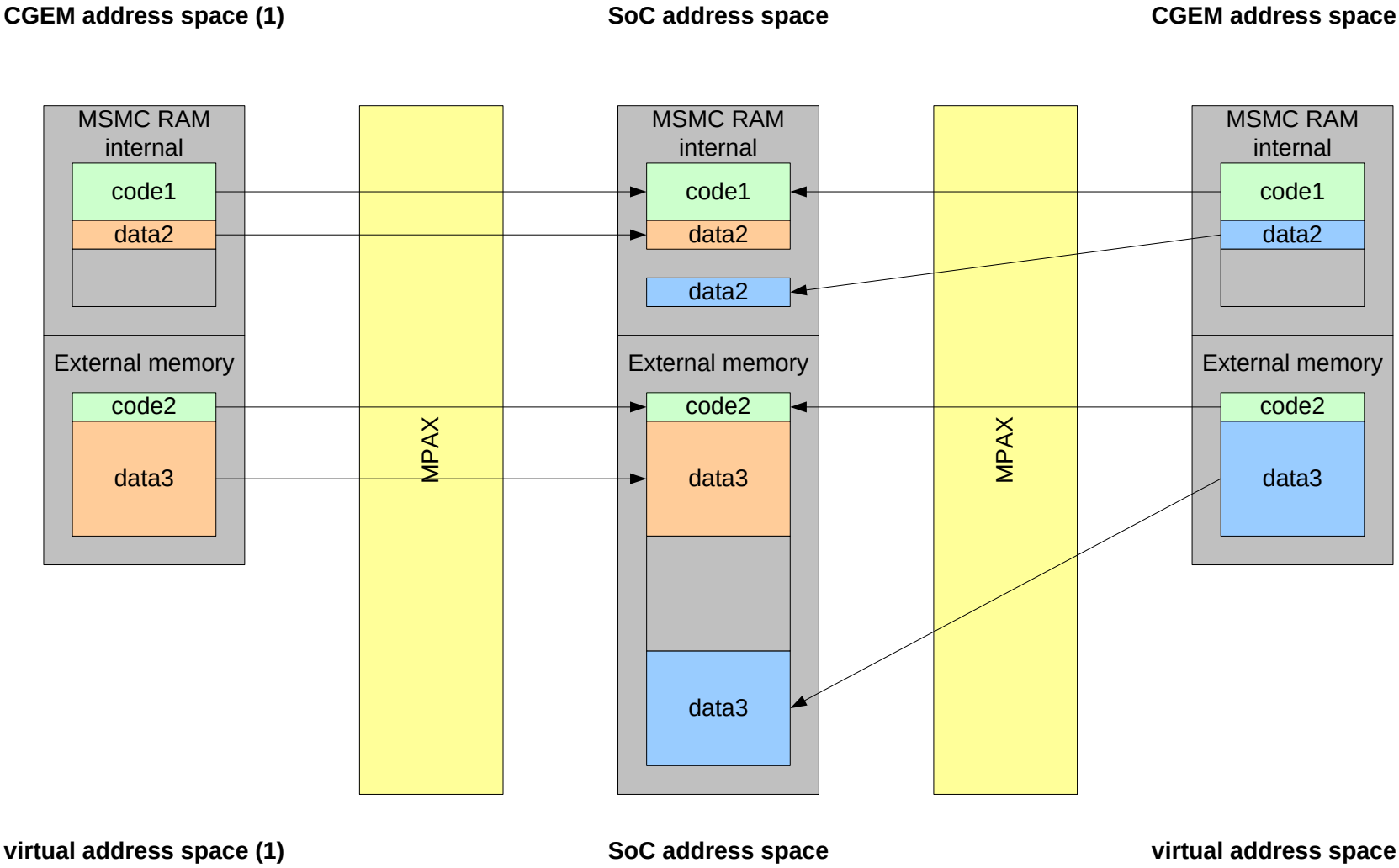
- C6678 DSP Overview
- Multi-core DSP programming
  - Memory Architecture Overview
  - Shannon Memory Architecture Improvement
  - Programming model
- Interconnection and resource sharing
- Peripherals overview

single program image

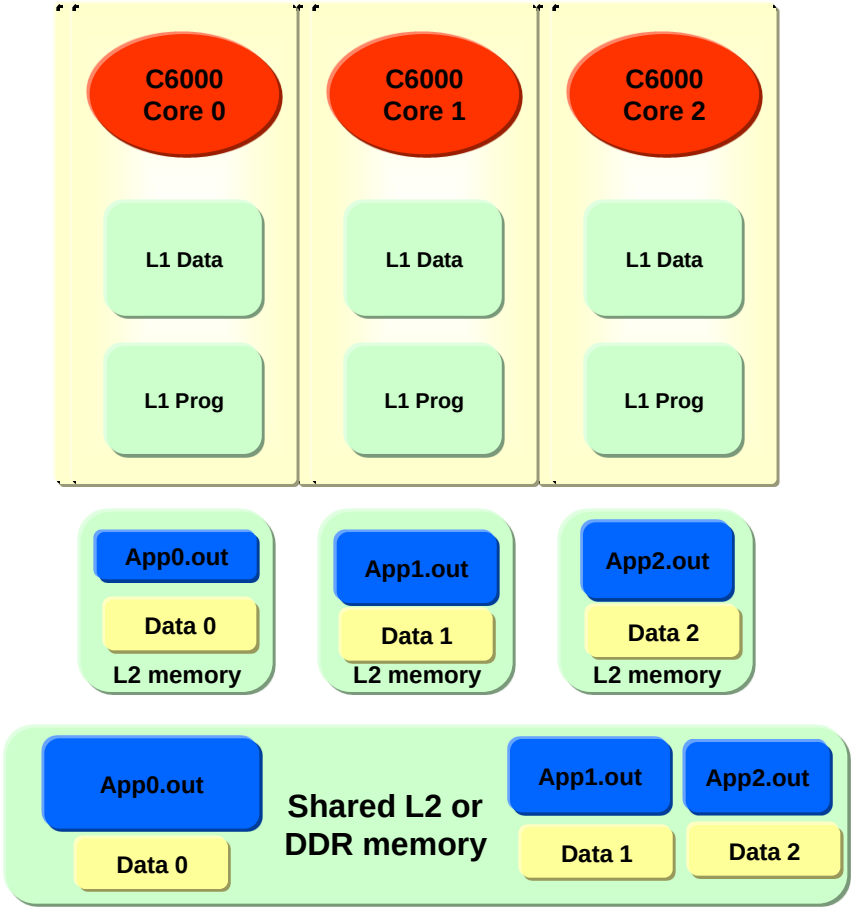


- Same image on each DSP core
- Aliased addressing used for DSP core to access local L2
- DNUM DSP core register for:
  - Global addressing when programming EDMA3, SRIO, …
  - Separate buffer per DSP core in DDR: dp= bufBase+ BUF_SIZE*DNUM

Shannon MPAX enables easy single program image



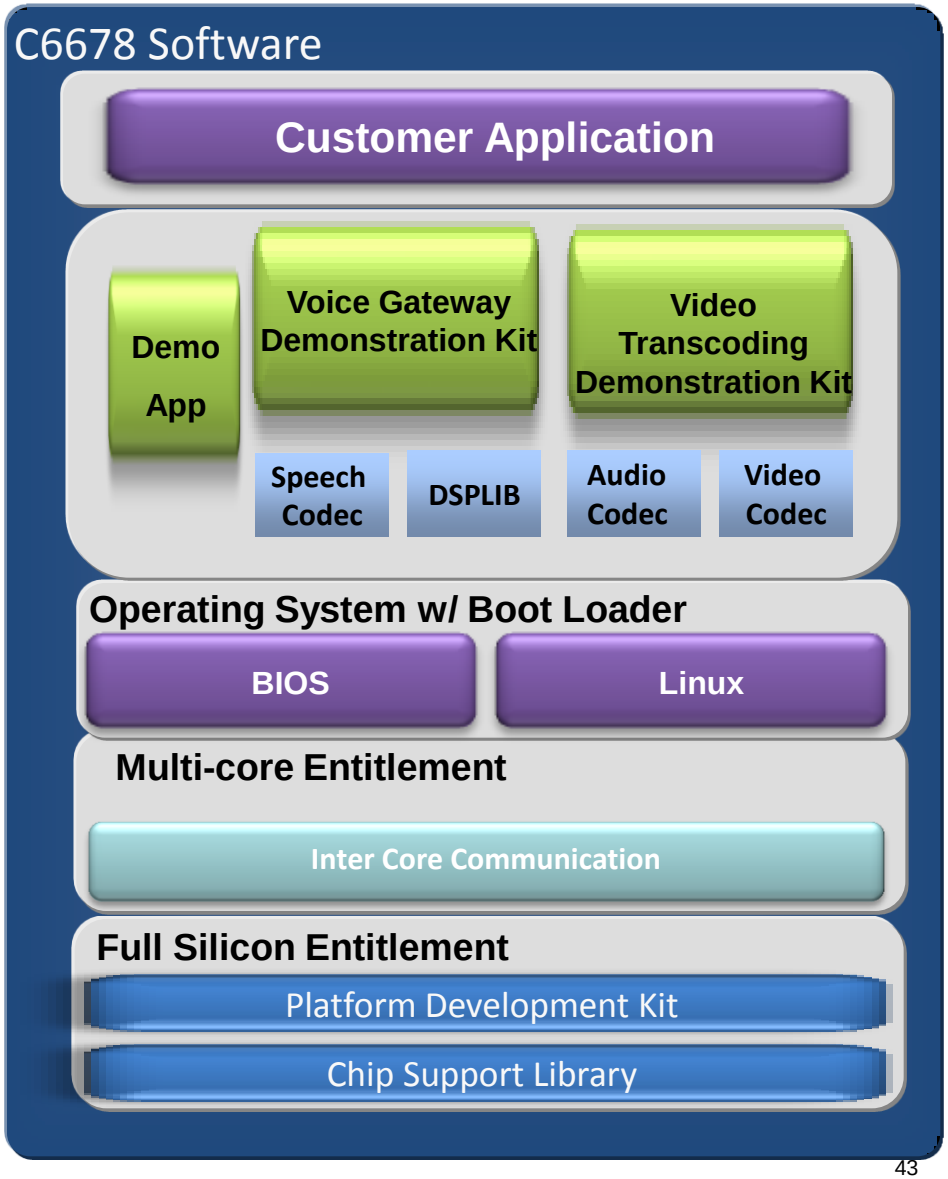
multiple program image



- Each DSP core has its image
- Static split of DDR2 per DSP core
- Global or local addressing used for L2 addressing

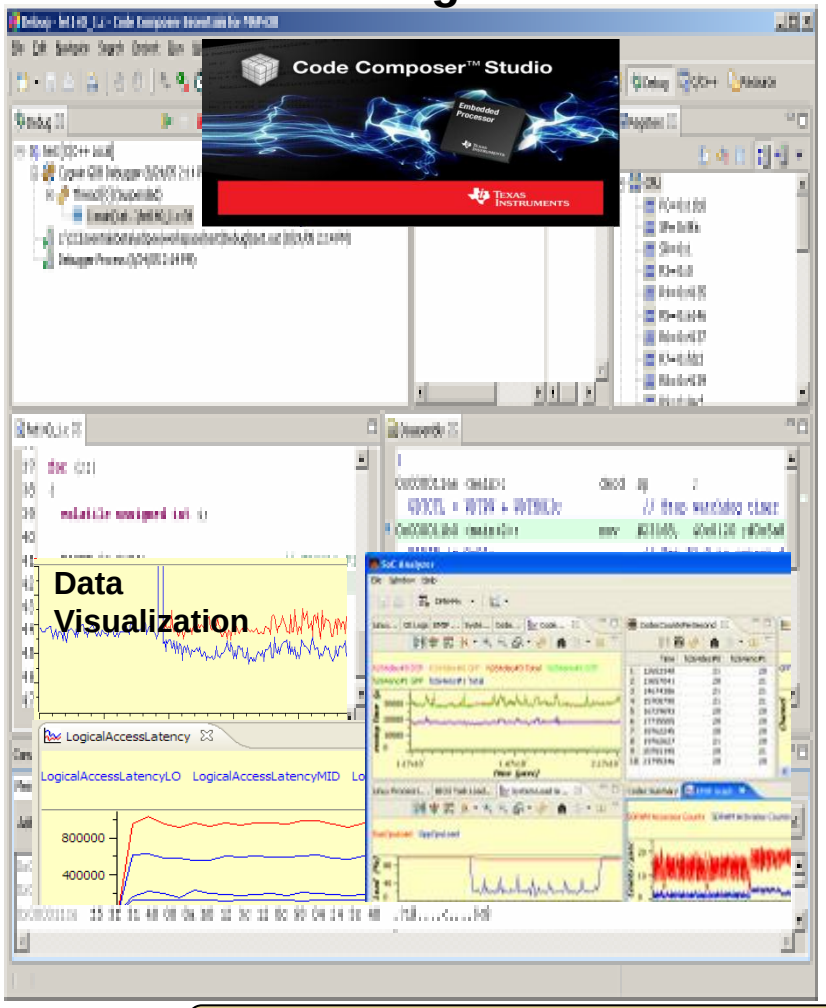
Shannon Software

- **Flexible development environment for the customer.**
- **Customer can choose to develop their application using all or any one of the software layers.**
- **Will contain following software layers**
  - **BIOS and Linux Operating System support**
  - **Chip Support Library**
  - **Platform Development Kit**
  - **Inter Core Communication**
  - **Optimized DSP functions library**
  - **Optimized Audio, Video and Speech codecs**
  - **Voice Gateway Demonstration Kit**
  - **Video Transcoding Demonstration Kit**
  - **Demonstration applications**

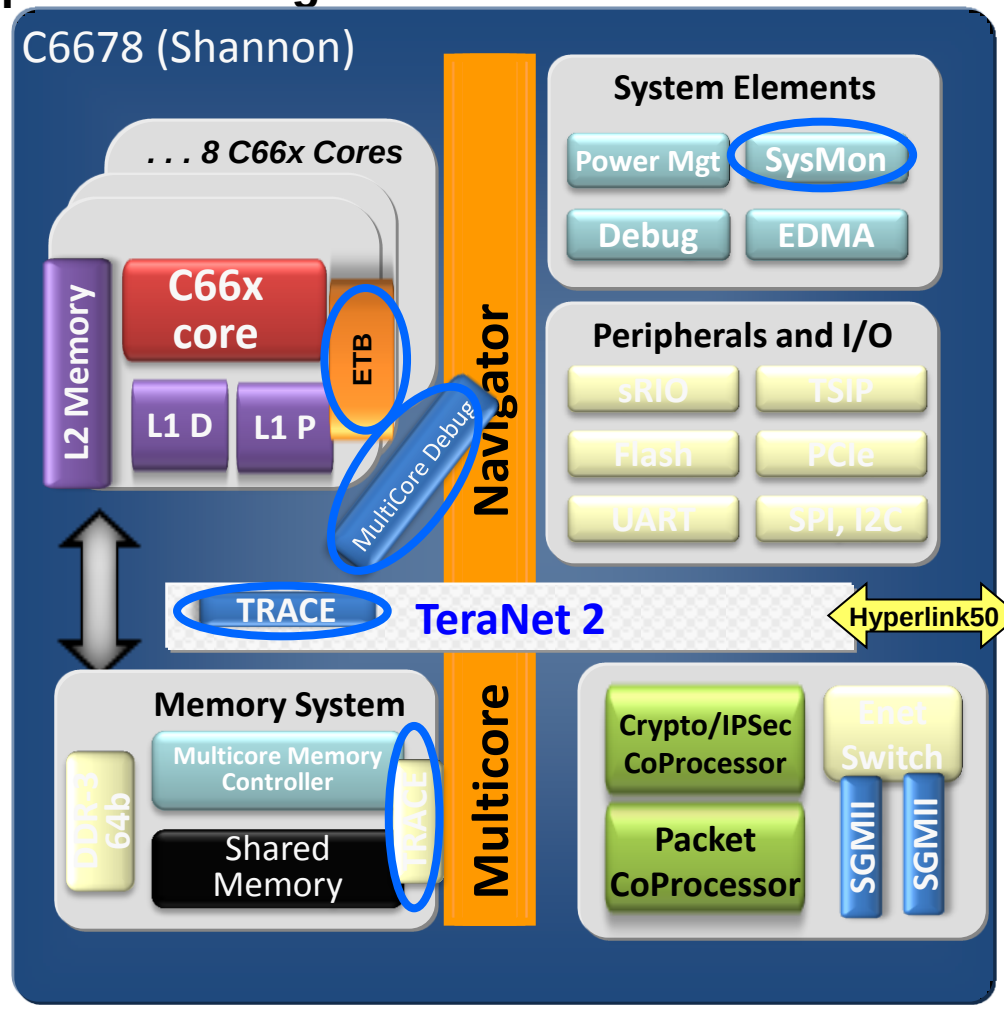


Shannon Debug

Best Multicore Debug and Visualization



Debug enabled Multicore SoC

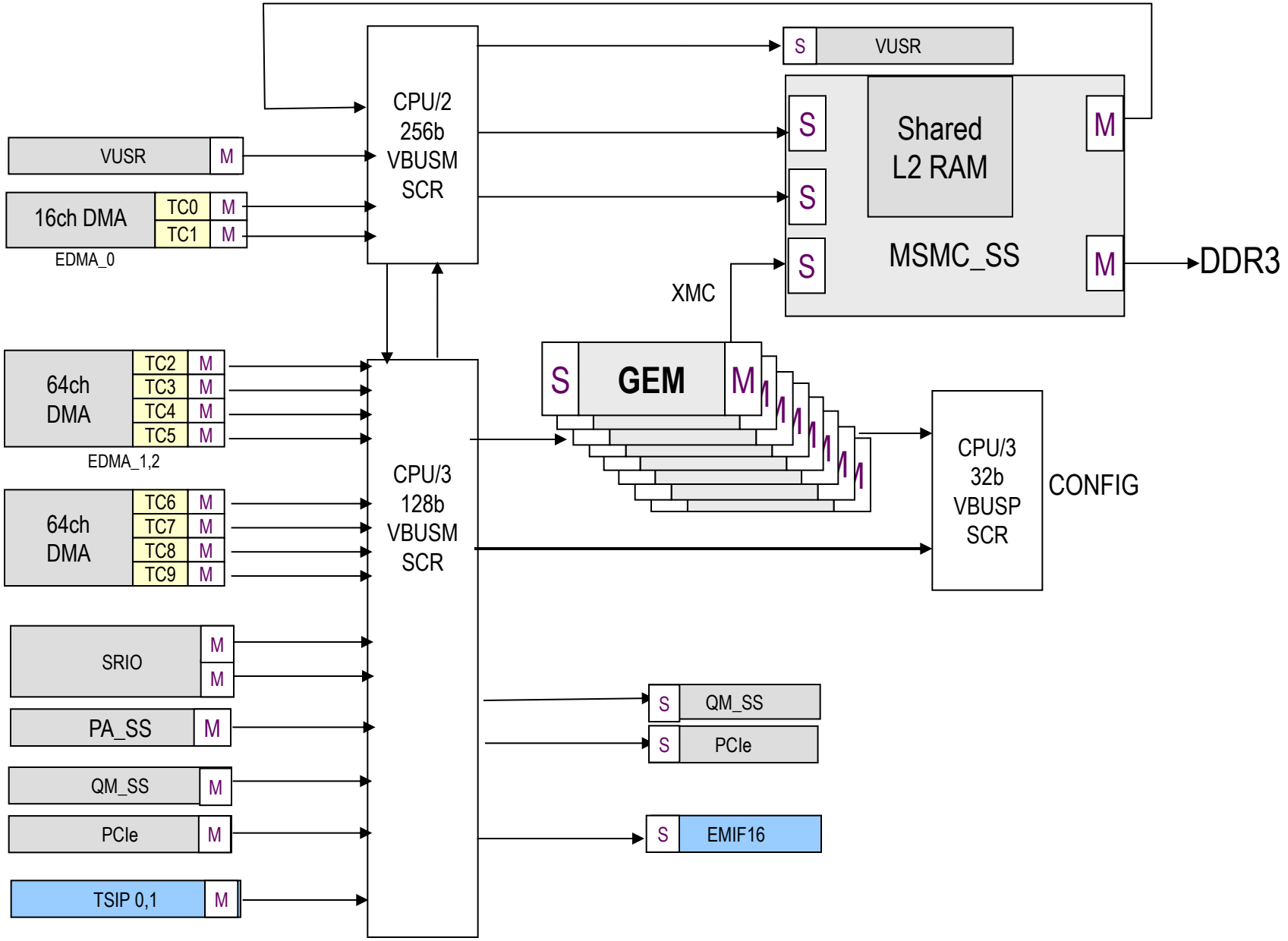


Debug visibility at core, across multicore and for SoC

Outline

- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
  - Interconnection Architecture
  - Shannon Hardware queue
  - Inter-core communication
  - Shared Resource Management
- Peripherals overview

Shannon Switch Fabric



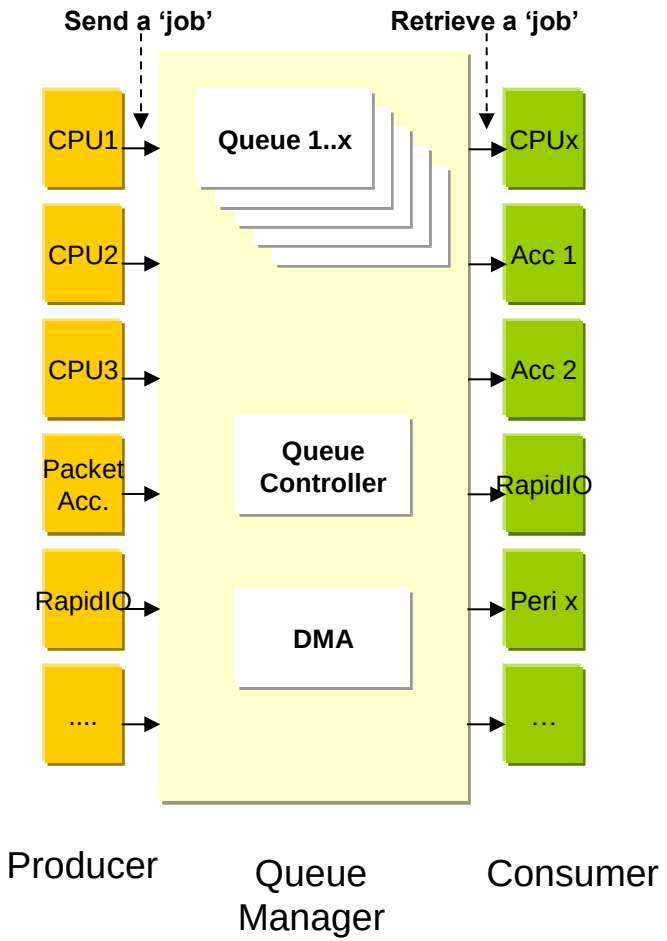
Outline

- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
  - Interconnection Architecture
  - Shannon Hardware queue
  - Inter-core communication
  - Shared Resource Management
- Peripherals overview

Hardware Queue Architecture

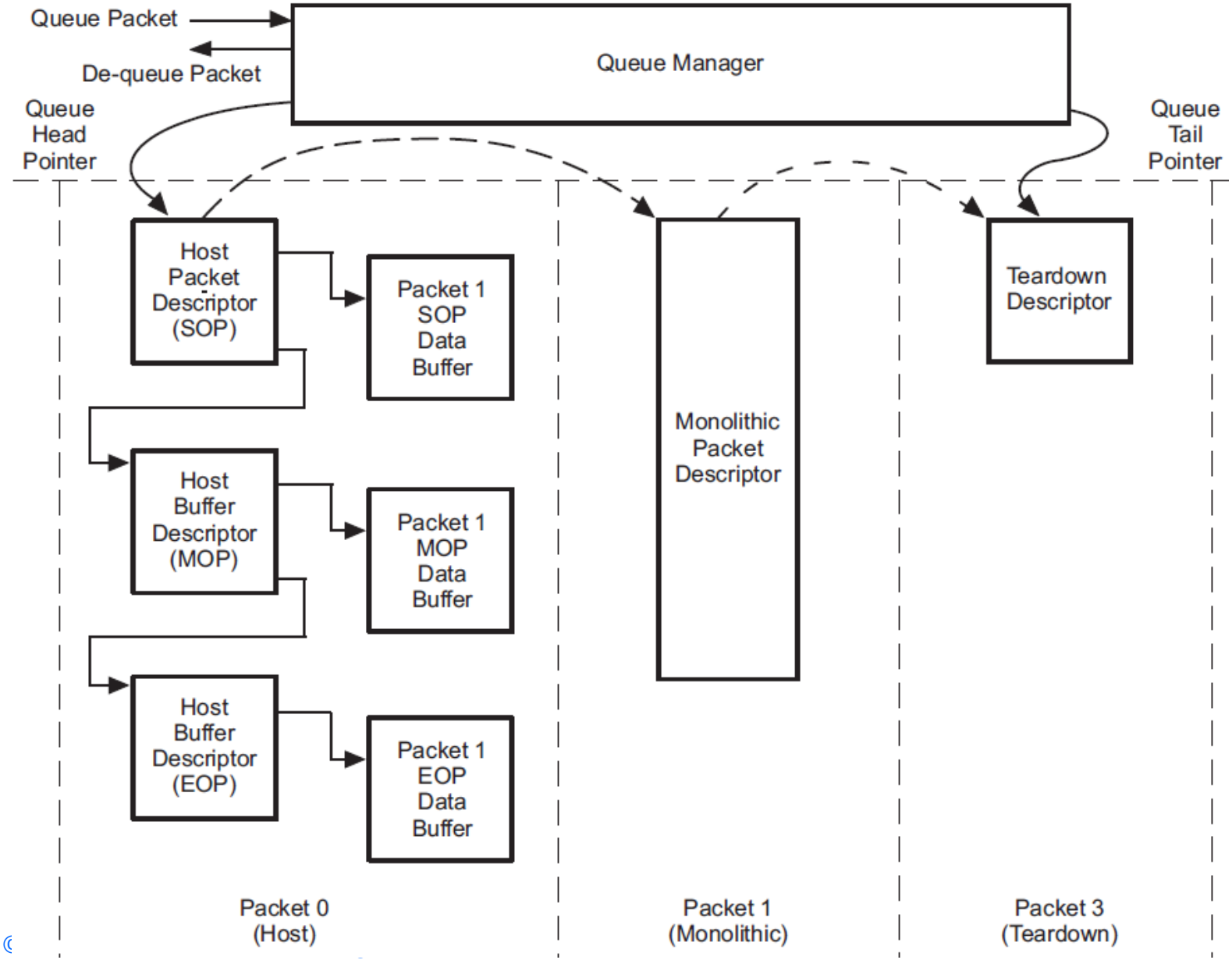
- packetized Data transfer architecture designed to minimize DSP core interaction while maximizing memory and bus efficiency
- the key communication platform for TI's future Infrastructure DSPs
- Used by following peripherals in Shannon:
  - Serial RapidIO, Packet Accelerator
  - Each module contains its own DMA to transfer associated data with the 'jobs', No CPU resources involved

Hardware Queue



- Producer writes 'jobs' into a Queue.
- Consumer reads 'jobs' from the Queue
- Supports Multiple In – Multiple Out
  - Multiple Producers can write to the same Queue
    ⇨ Used to share common Hardware
  - Multiple Consumers can read from the same Queue
    ⇨ Used for Load Balancing
- Abstracts the Consumer
  - Consumer can be a Hardware IP (accelerator, peripheral) or a software (ie a CPU core)
  - Transparent for the Producer
  - ⇨ 'Easy' to upgrade to new hardware. The 'job gets done'.
  - ⇨ Minimize changes to Host software, Easy maintenance

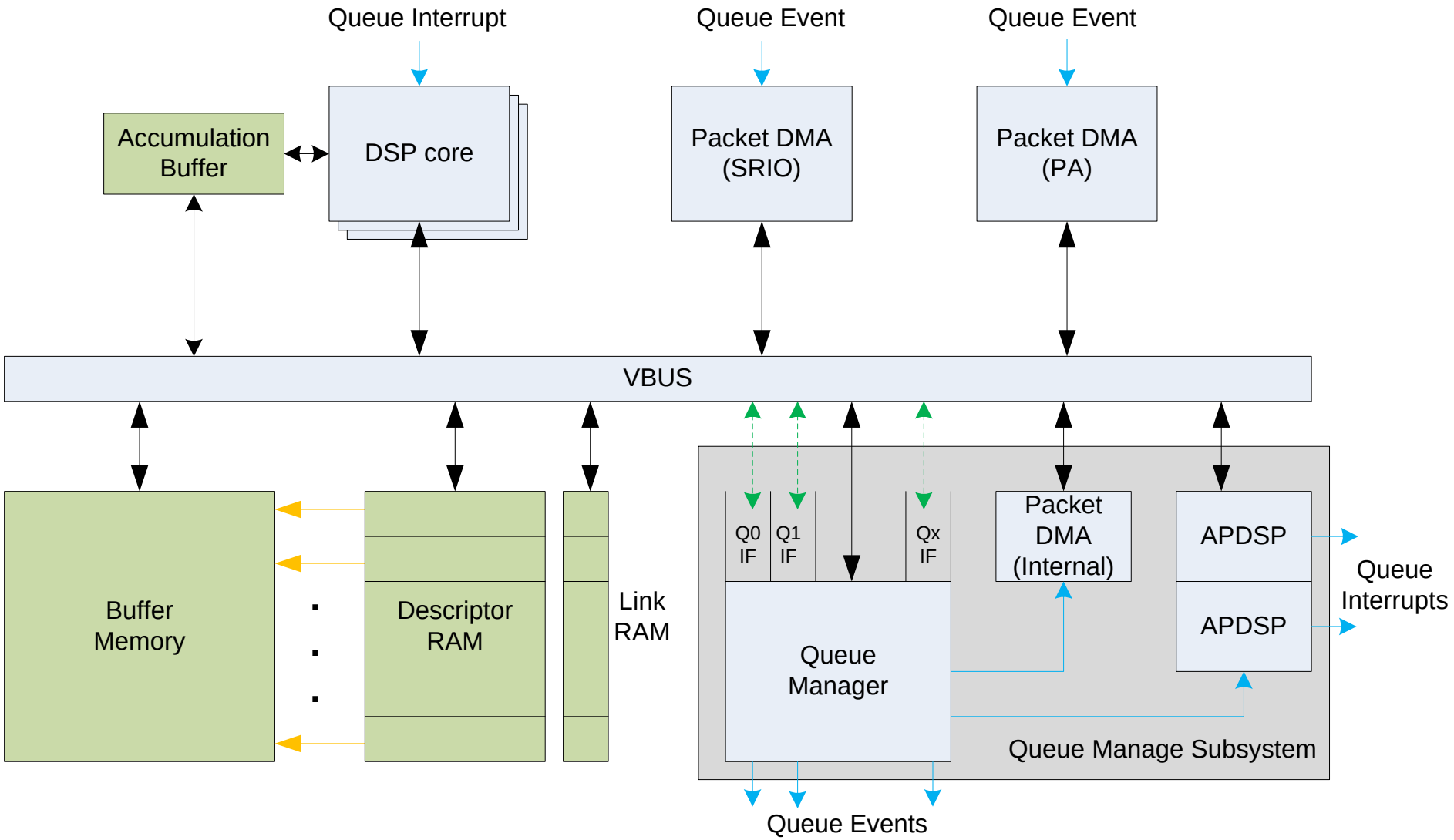
Packet Queuing Data Structure Diagram



Hardware Queue Operation

- Push to a queue
  - Host write pointer of new descriptor to a queue register.
  - Queue manager links (modify the link RAM) the new descriptor to the tail (or header) of the queue.
  - Tail (or header) pointer points to the new descriptor.
- Pop from a queue
  - Host read a descriptor pointer from a queue register.
  - Queue manager returns the descriptor pointed by the header pointer
  - Header pointer points to the next descriptor.
- Monitor queue
  - Queue manager generates events when queue changes: not empty, entry count, exceed threshold, starvation…
- Queue Diversion
  - Entire queue contents can be cleared or moved to another queue destination using a single register write

Shannon Hardware queue architecture

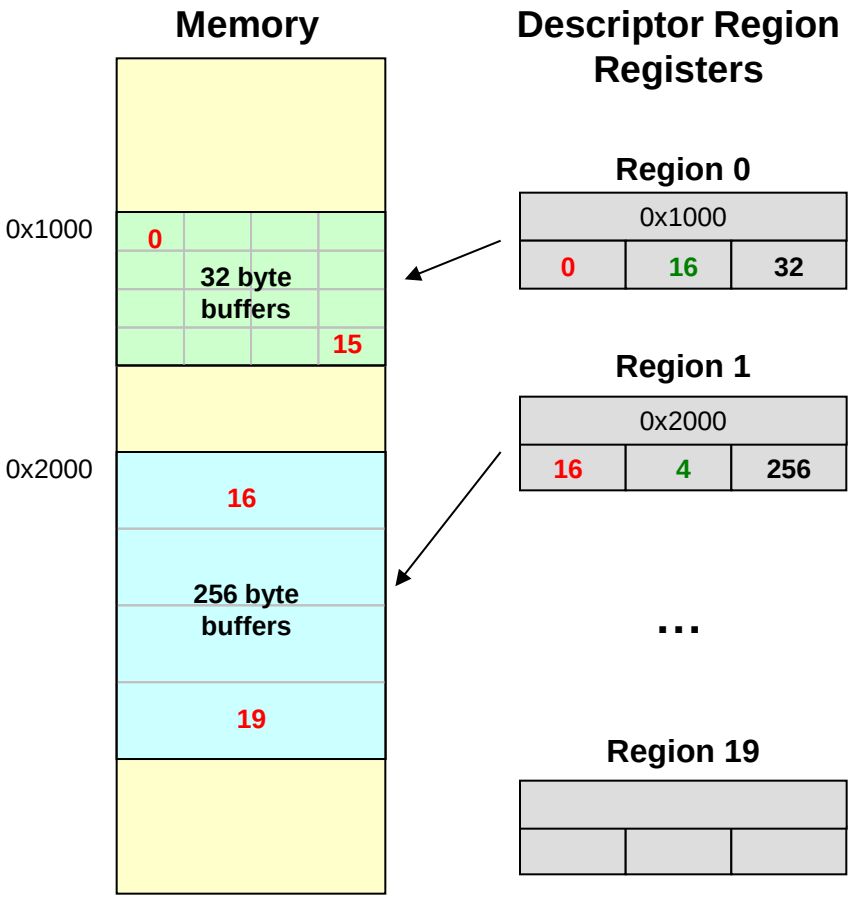


Queue Manager Subsystem

- Support 8192 queues
- HW queues are multi-core safe without mutual exclusion, multiple senders can use a destination queue without restrictions
- Can Notify Packet DMA when transfer is pending
- Can notify DSP core when packet is pending, can copy descriptor pointers of transferred data to destination core's local memory to reduce access latency
- Internal Packet DMA
  - Transfer packet from one queue to another queue. Good for core to core data transfer.

Descriptor RAM

- Data elements (buffers) to be passed on queues are first described to a descriptor region manager built into the QM.
- Although technically called descriptors, these memory elements can hold any arbitrary data.
  The size of the data elements must be a power of 2, from 32 bytes to 8192 bytes in length.
- 20 configurable memory regions (for descriptor storage)
- The number of elements in the region must be a power of 2, from 32 buffers to 4096 buffers in the region.



Linking RAM

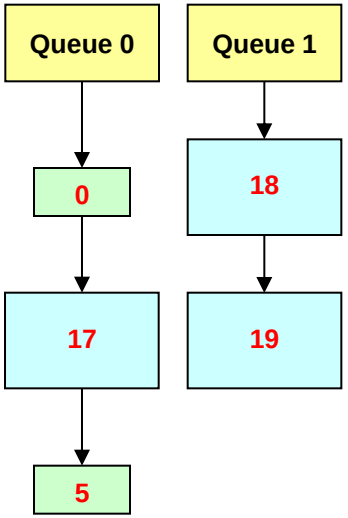
- Linking RAM contains 1 entry for each descriptor . Linking RAM entry is effectively an extension of the descriptor
- Linking RAM stores Forward data pointer that is critical for the PUSH / POP operations performed by the Queue Manager
- Linkage between physical address of descriptor and physical address of Linking RAM is performed inside the QM using information provided in the Queue Management configuration registers
- Linking RAM is typically placed in local memory for speed. This allows data elements to be linked and unlinked in a queue very quickly, even though the buffers themselves may be in external memory
- There is no limit to the length of a single queue, only a limit on the total number of data elements in the system.
- 2 configurable Linking RAM regions

Linking RAM

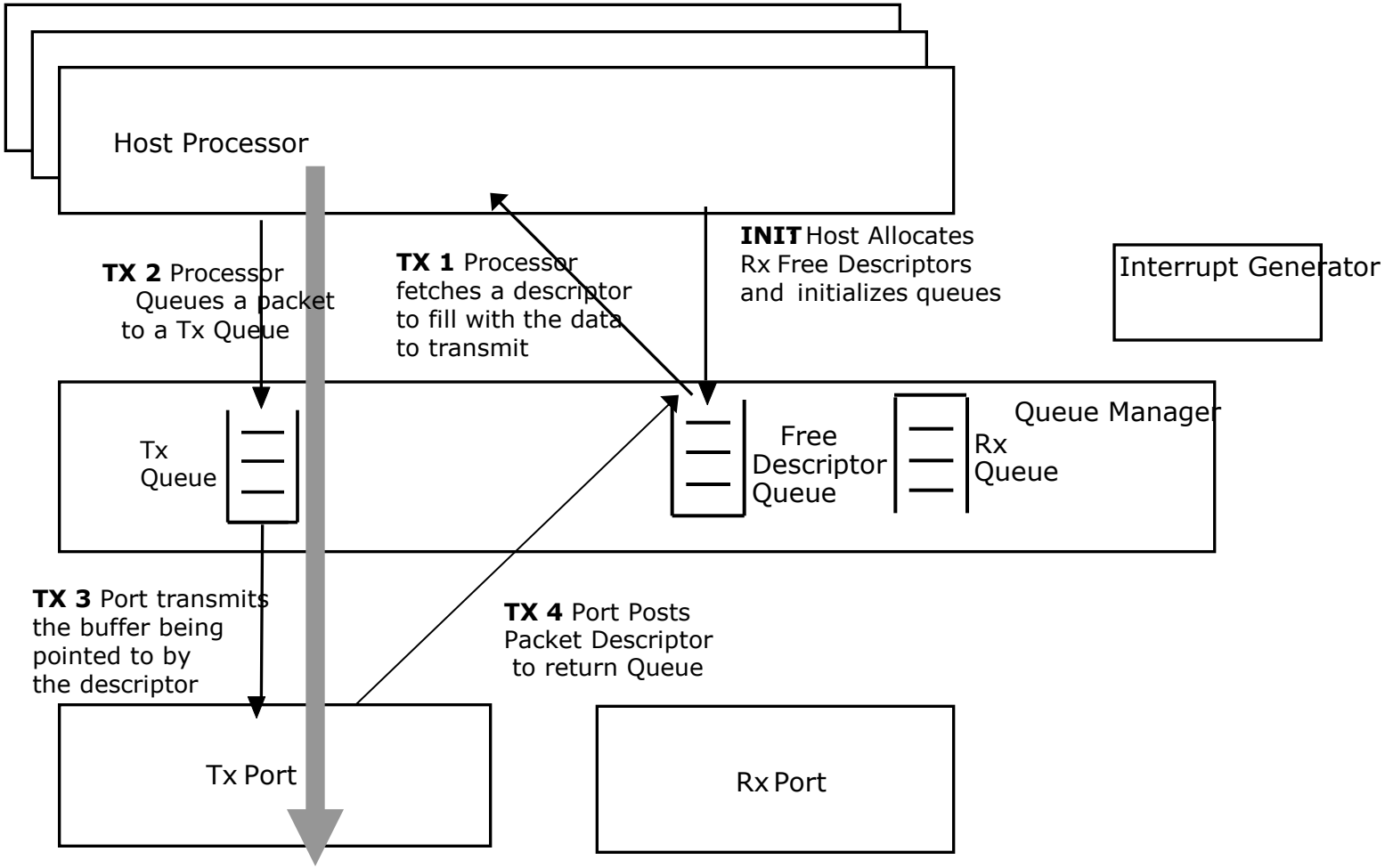
Forward Pointer Table

17	-	-	-
-	x	-	-
-	-	-	-
-	-	-	-
-	5	19	x

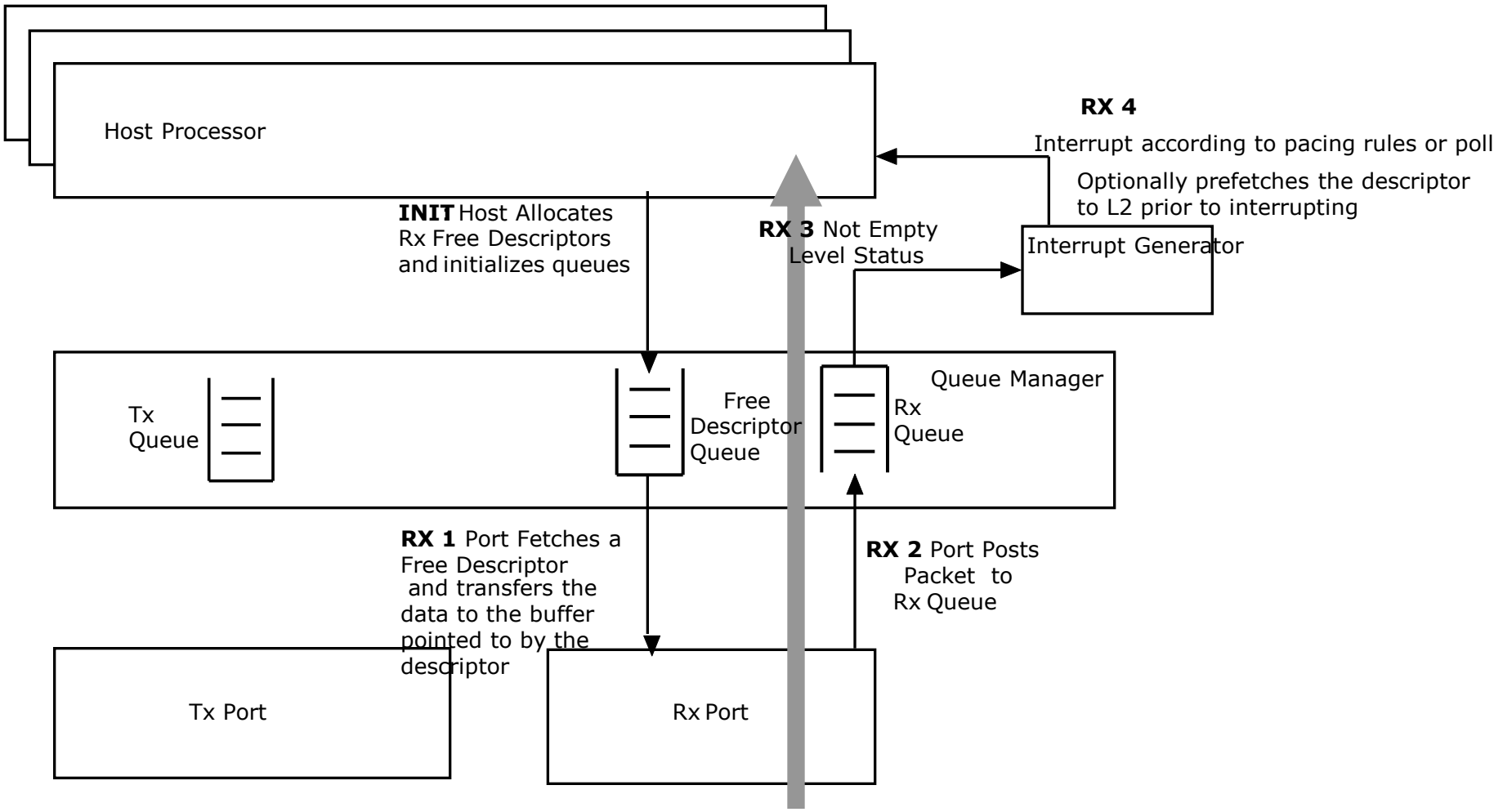
Queue Contents



Queue Data Flow Example, Transmit

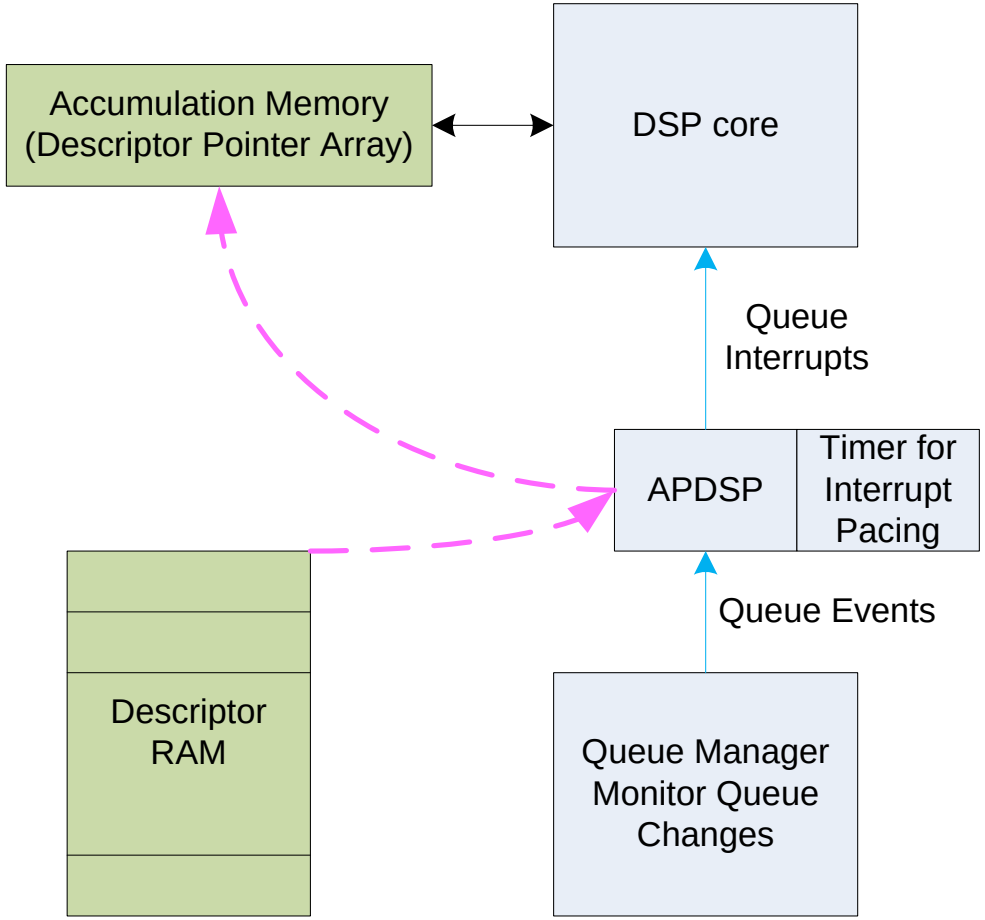


Queue Data Flow Example, Receive



Accumulator (A Programmable DSP)

- Accumulator is used to help DSP core efficiently POP descriptor pointers from queue.
- Accumulator pop descriptor pointer from queue and write to accumulation memory (normally in DSP local memory).
- Accumulator generates interrupt to DSP core according to interrupt pacing configuration.
- Two Accumulator (PDSP)
  - One generate 32 interrupts, each for one queue.
  - The other generate 16 interrupts, each is combined event for 32 queues. Totally monitor 16x32 queues.



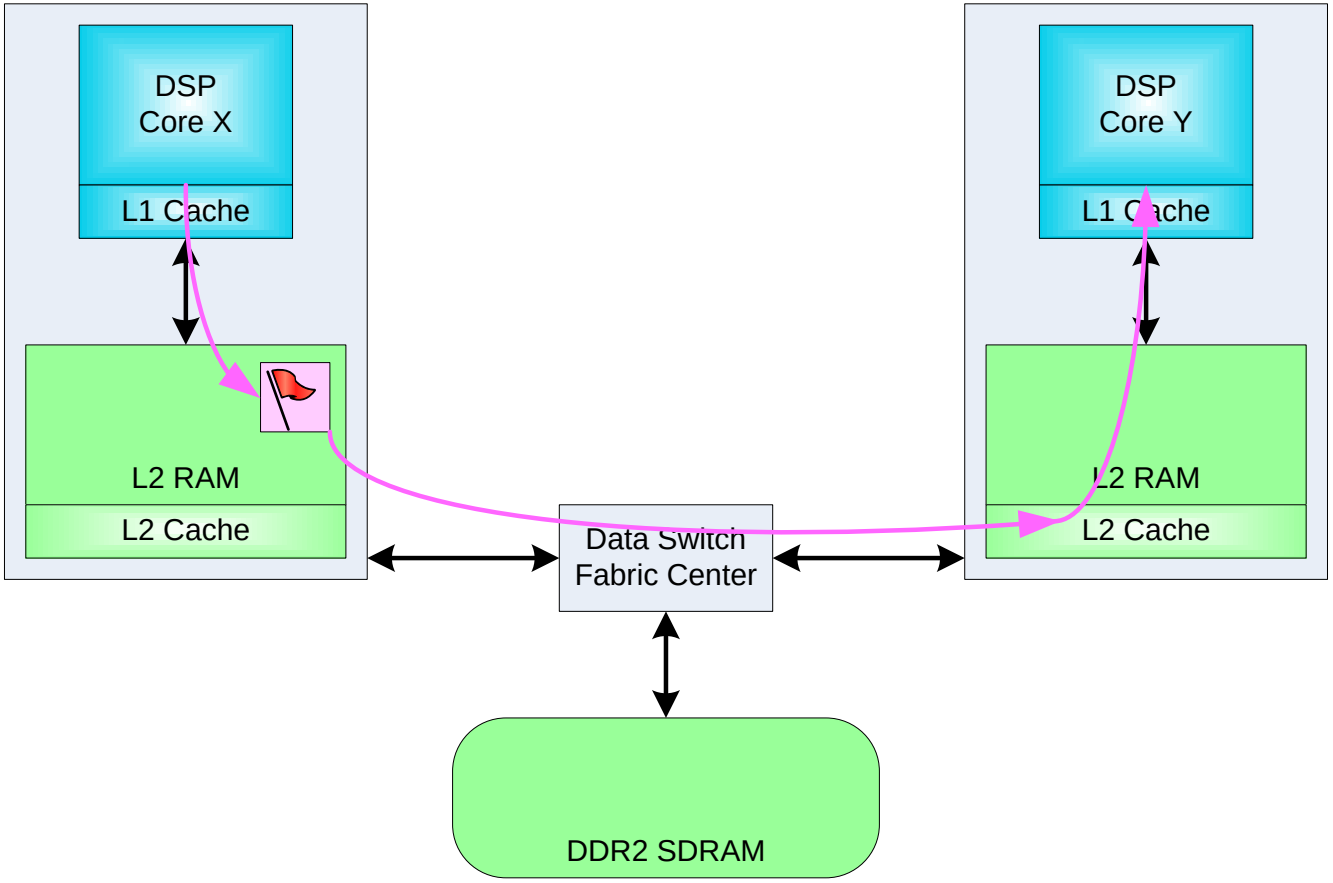
Hardware queue Performance Consideration

- **Push Operation**
  - **1~4 words write. Since it is post operation, normally, do not stall DSP core.**
- **Pop Operation**
  - **1~4 words read. Stall DSP core about 80~100 cycles.**
  - **Accumulator (PDSP) can pop the descriptors to DSP local memory which will save DSP cycles dramatically.**
- **Descriptor Access**
  - **Write/read full descriptor may consume many cycles.**
  - **For most applications, DSP core can initialize all descriptors during initialization, and only write/read few fields of the descriptor during run time.**

Outline

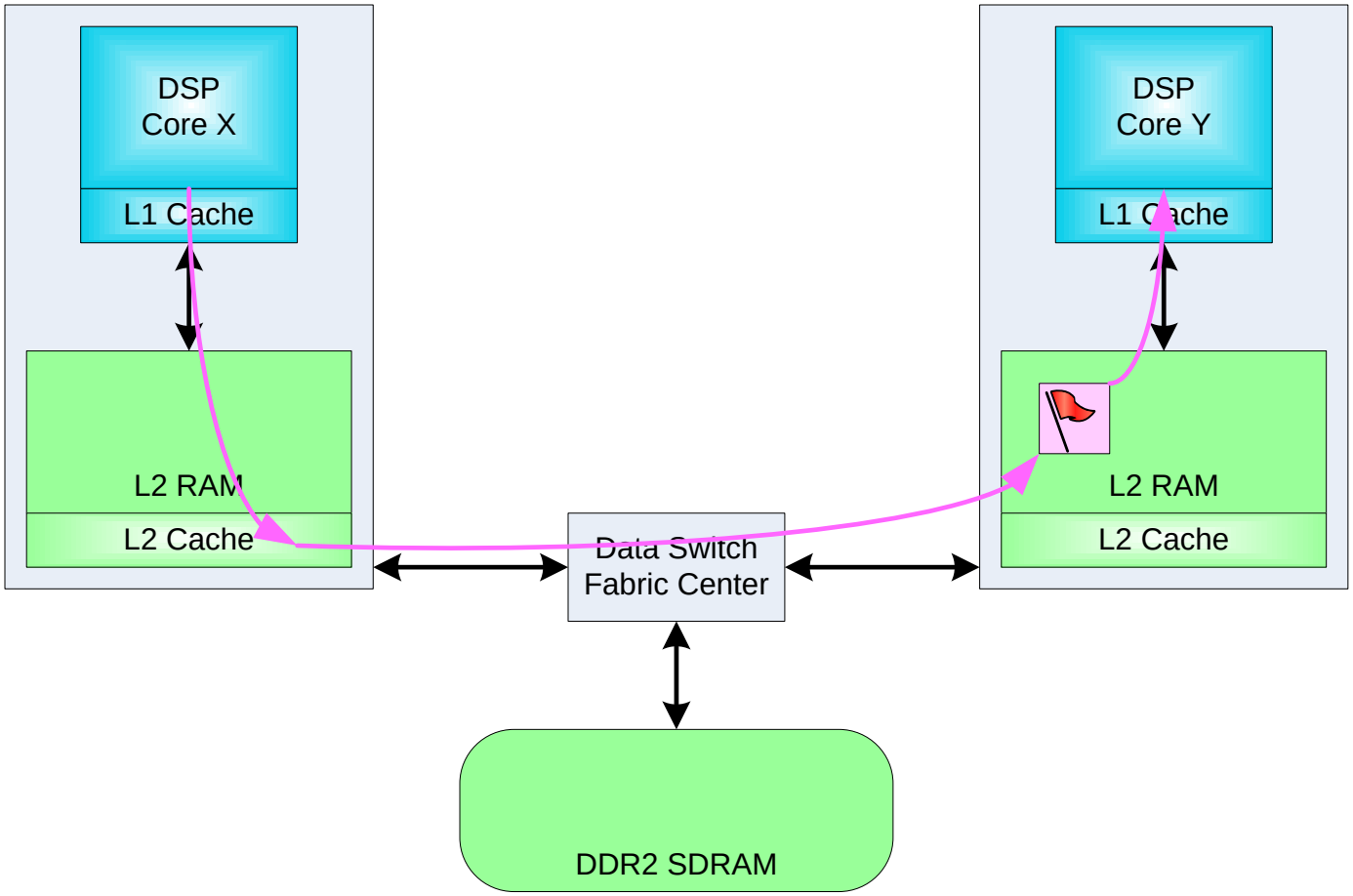
- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
  - Interconnection Architecture
  - Shannon Hardware queue
  - Inter-core communication
  - Shared Resource Management
- Peripherals overview

Shared Data in the L2 SRAM of transmitter



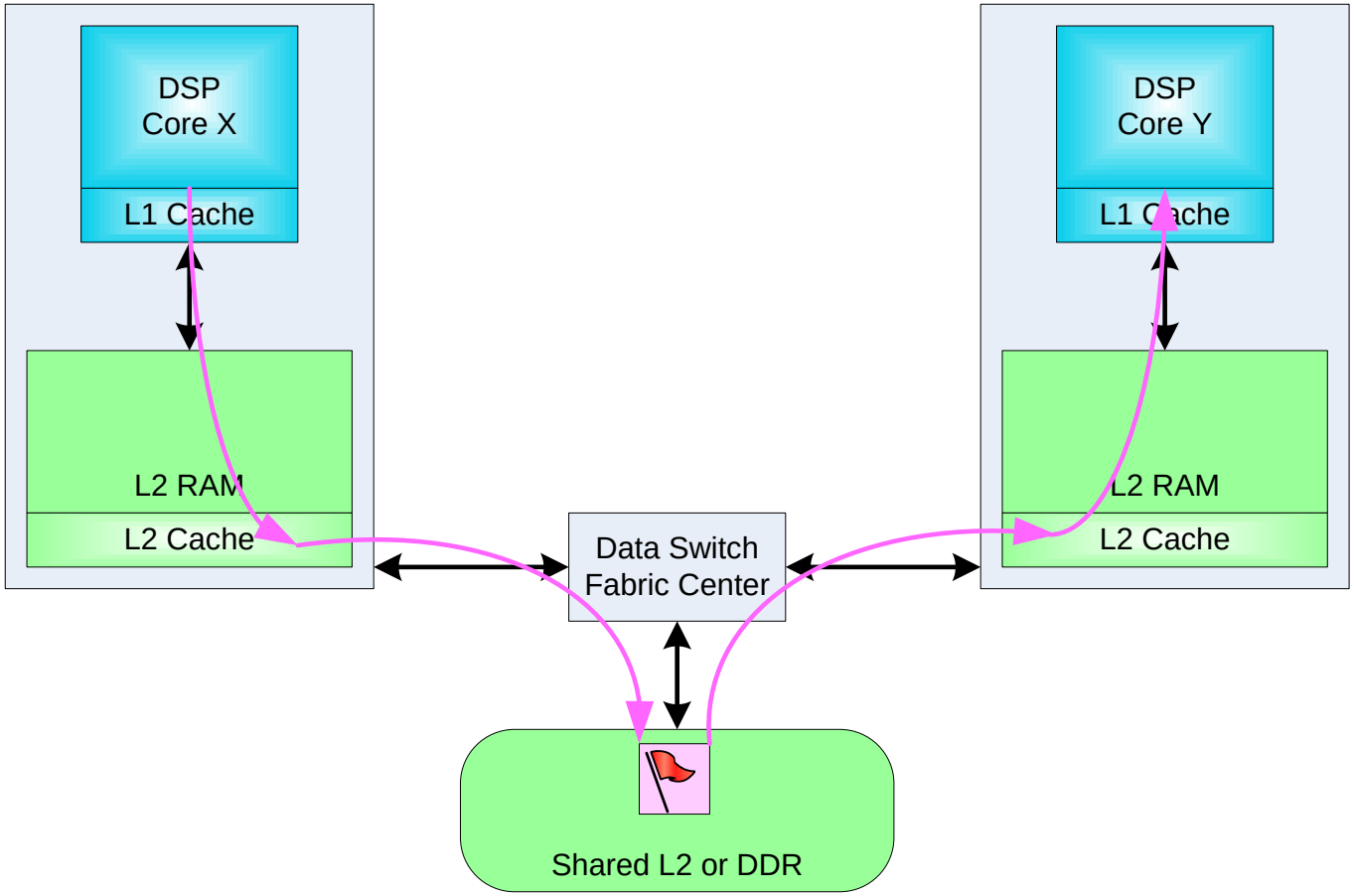
- **If cache is enabled, Core Y needs invalidate cache before read**

Shared Data in the L2 SRAM of receiver



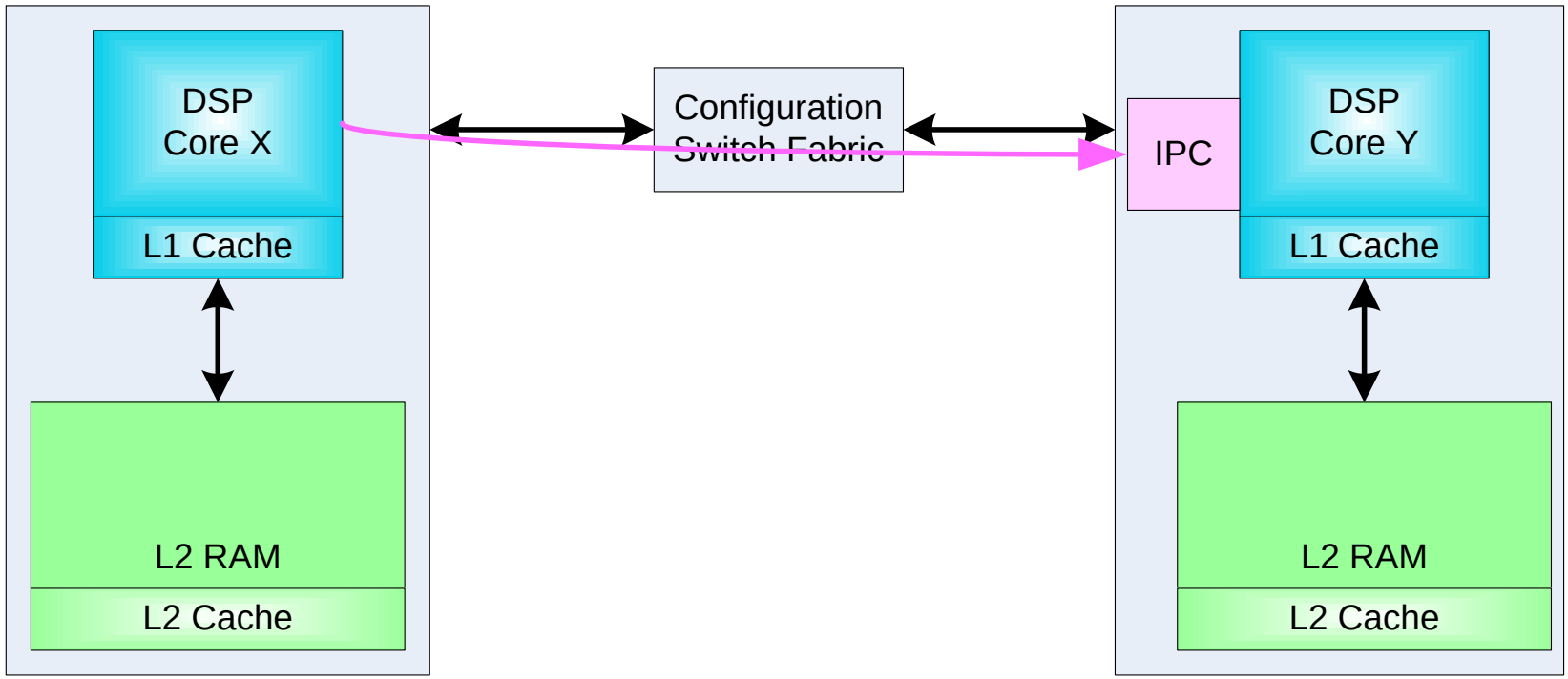
- **If cache is enabled, Core X needs write back cache after write**

Shared Data in the shared memory



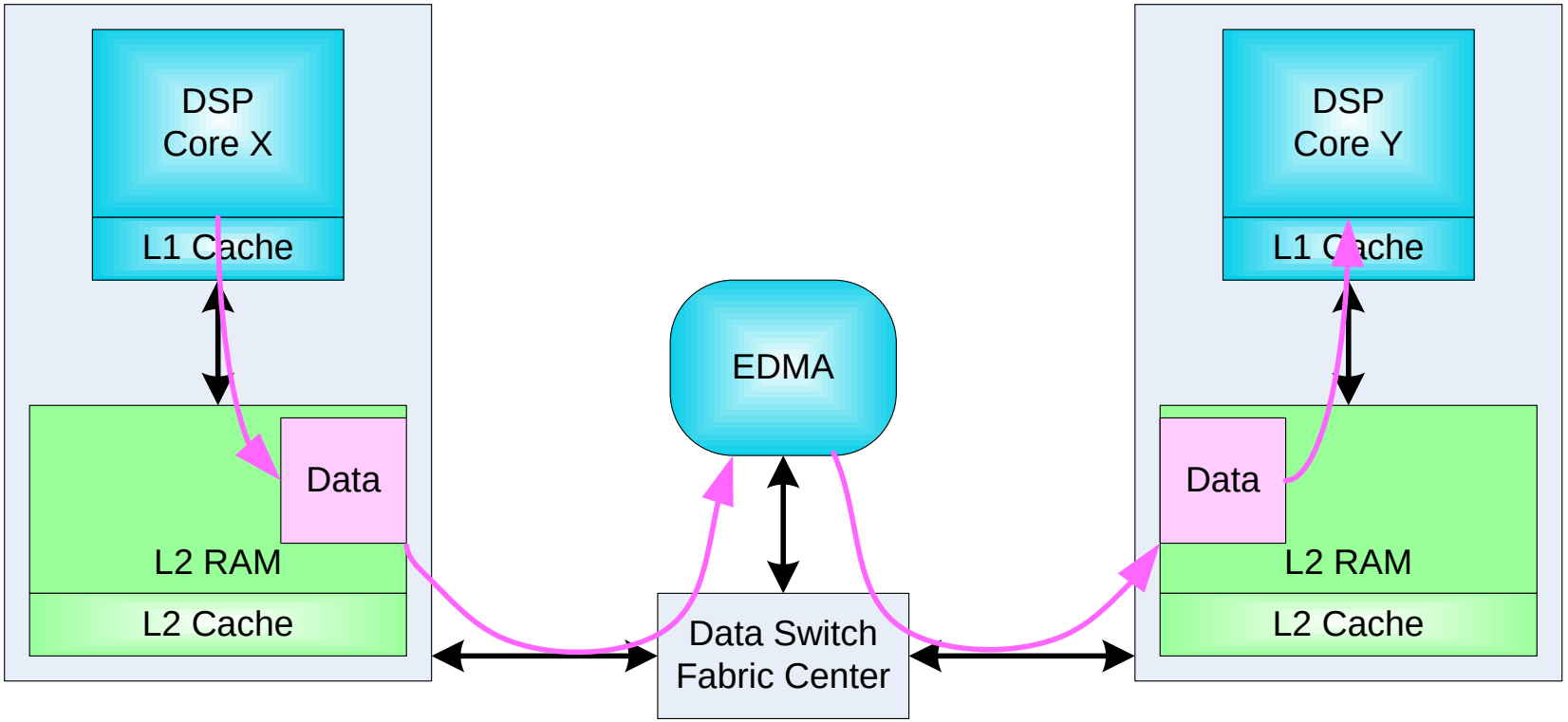
- **If cache is enabled, Core X needs write back cache after write; core Y needs invalidate cache before read**

Use IPC register for inter-core communication



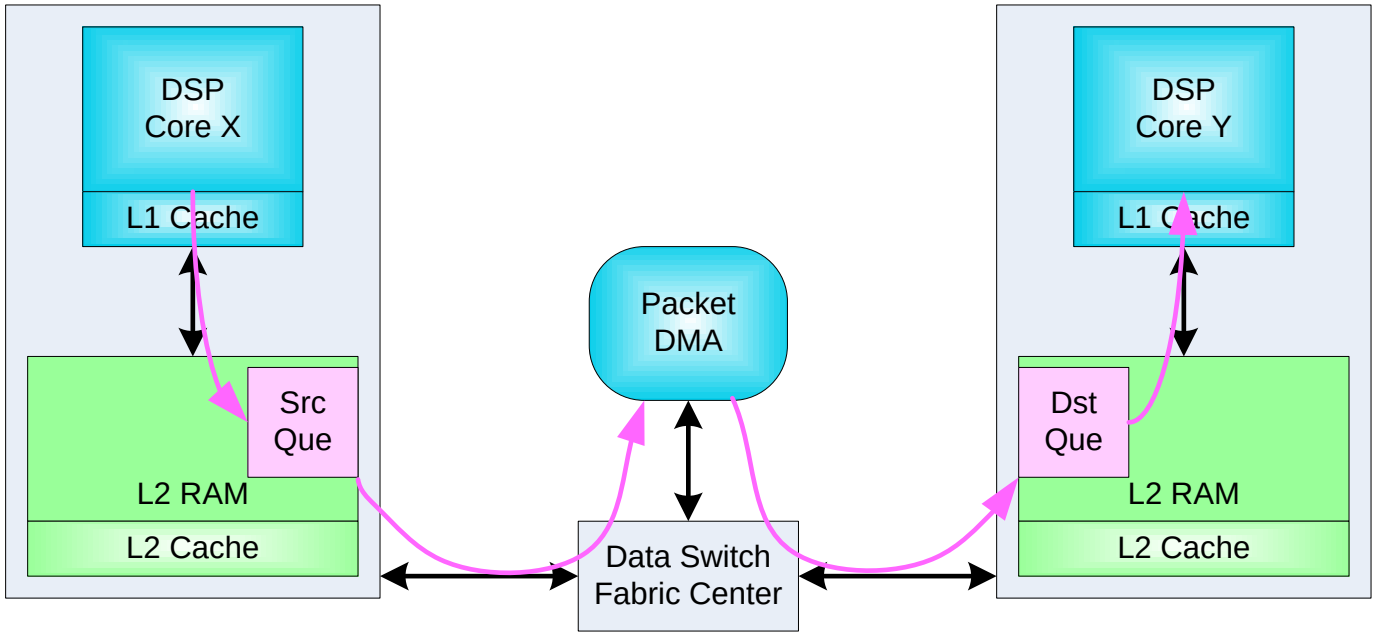
- **Interrupt is generated for Core Y**
- **No cache coherency issue**

Inter-core Data Block exchange with EDMA



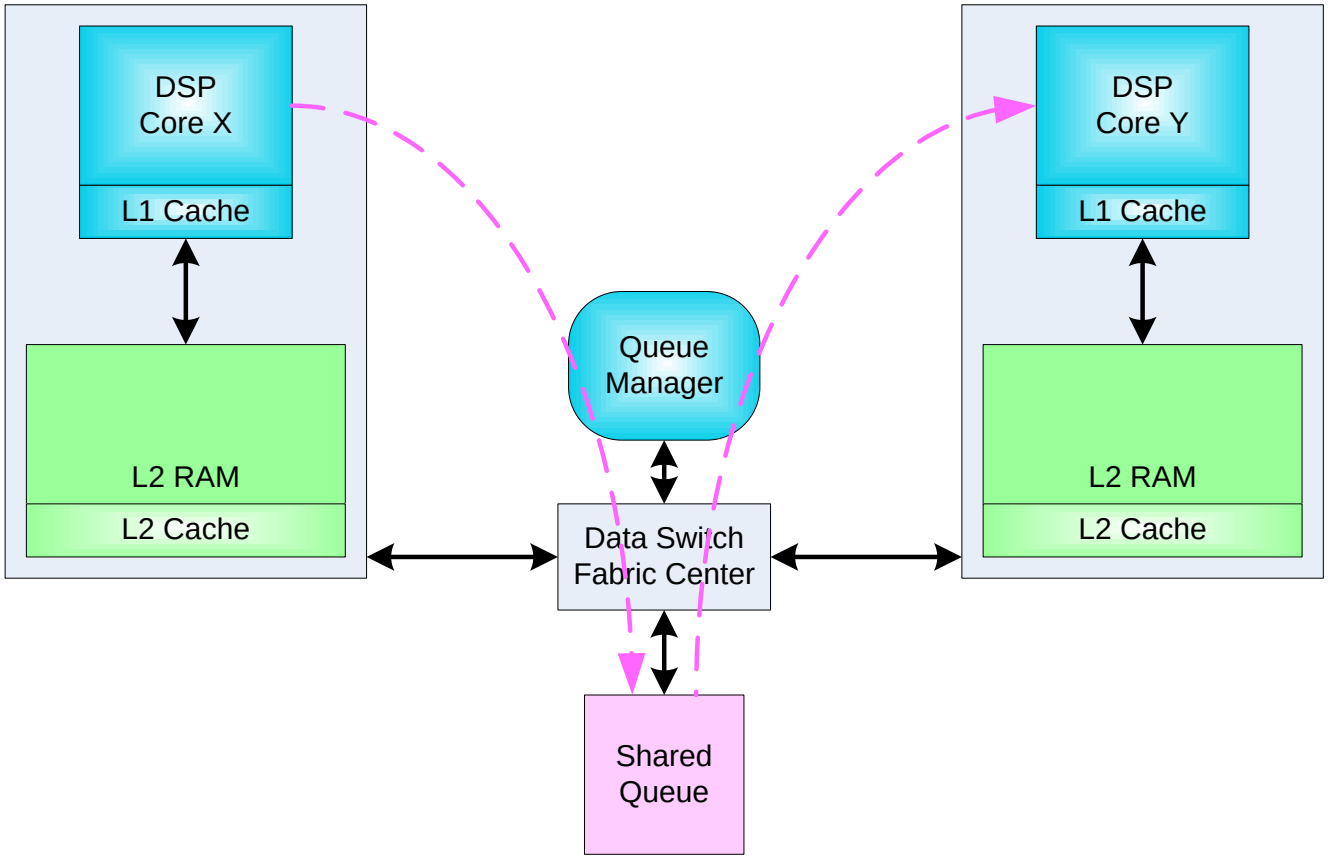
- **Interrupt is generated for Core Y**
- **No cache coherency issue**

Inter-core data exchange through hardware queue (Packet DMA copy)



- **Core X simply push data to Source Queue**
- **Packet DMA transfer the data Dest Queue**
- **Core Y simply pop data from Dest Queue**
- **If Queue buffers are in L2 RAM, Software on both cores do not need maintenance the cache coherency.**

Inter-core data exchange through hardware queue (Zero Copy)



- **Core X push data to Shared Queue, Core Y pop data from Shared Queue**
- **Multi-core can access Shared Queue simultaneously without mutual exclusion**
- **Software need maintenance the cache coherency.**

Outline

- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
  - Interconnection Architecture
  - Shannon Hardware queue
  - Inter-core communication
  - Shared Resource Management
- Peripherals overview

Shared resources

- Internal shared L2 and External Shared memory (DDR)
  - Each core access shared memory independently. Arbitration handled by switch fabric and end-point arbiters.
- Shared on-chip Peripherals
  - Configuration: Typically done at startup to set the operating mode of a particular logic block (e.g. DDR settings). Should be done by a single core as part of the boot process.
  - Use:
    - Peripherals with Hardware queue, Each core access hardware queue independently. Arbitration handled by queue manager.
      - Ethernet, SRIO on Shannon…
    - Multi-channel peripherals can be split amongst the cores for concurrent, orthogonal control
      - EDMA, TSIP, Timer…
    - Single-channel peripherals can be controlled by a single master, servicing the other cores if needed. Or mutual exclusively used by multi-masters through semaphore.
      - I2C, SPI…

System-level prioritization for arbitration

- A user-specified priority may be assigned to:
  - Any DSP core accesses
  - Any EDMA, sRIO, Ethernet, … on-chip transfers
- Each of the master ports are assigned a priority (8 levels) configurable

Hardware Semaphores on Shannon for atomic accesses

- **What function does the Semaphore module provide?**
  - **A method to control who accesses a shared resource**
  - **Provides accesses for shared resources in an atomic manner**
    - **Read-modify-write sequence is not broken**
- **Features of the Semaphore module**
  - **Binary Semaphore**
  - **Contains 64 semaphores to be used within the system**
  - **Two methods of accessing a semaphore resource**
    - **Direct Access**
      - **A core directly accesses a semaphore resource. If free, the semaphore will be granted. If not, the semaphore is not granted**
      - **Useful if the system can afford to poll for the semaphore**
    - **Indirect access**
      - **A core indirectly accesses a semaphore resource by writing to it. Once it is free an interrupt will notify the DSP core that it is available.**

Outline

- C6678 DSP Overview
- Multi-core DSP programming
- Interconnection and resource sharing
- Peripherals overview

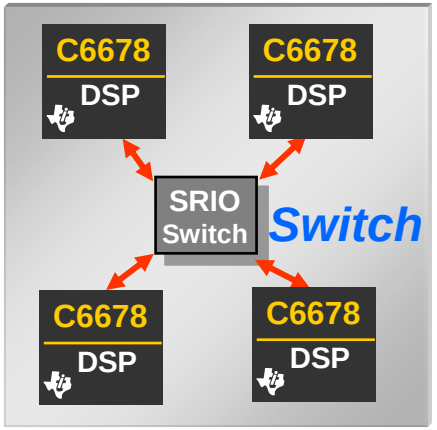
Shannon RapidIO Gen 2 Features and Enhancements

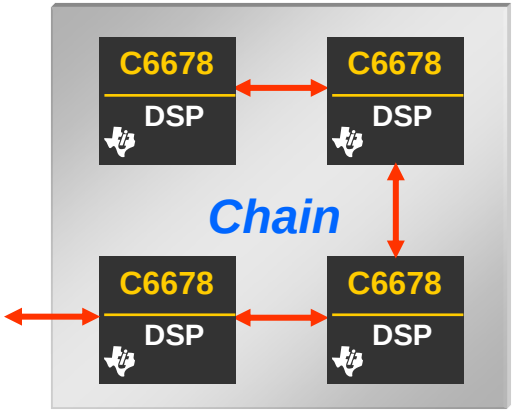
- 4 lanes – options include 2x

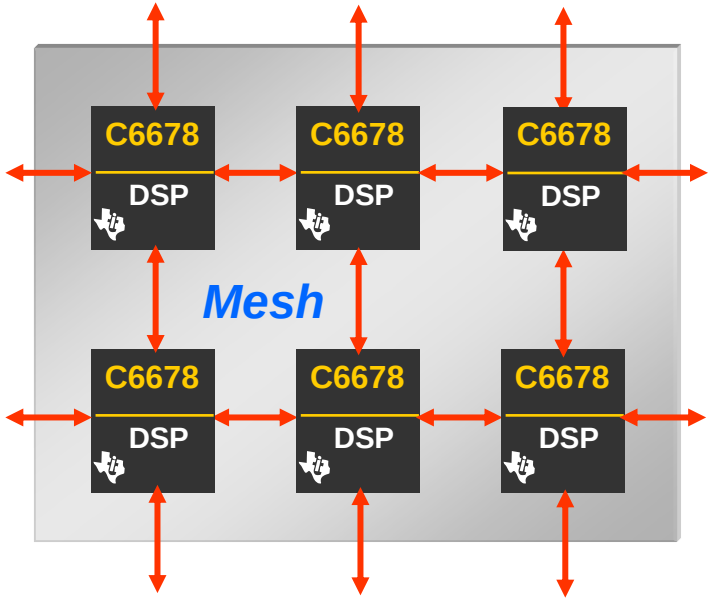
Mode 0	1x	1x	1x	1x
Mode 1	1x	1x	2x	
Mode 2	2x		1x	1x
Mode 3	2x		2x	
Mode 4	4x			

- Baud rates: 5 Gbaud per lane in addition to 1.25, 2.5, 3.125 Gbaud per lane
- DeviceID Support
  - 16 Local DeviceIDs (up from 1)
  - 8 Multicast IDs (up from 3)
- 24 Interrupt outputs (up from 8)
- Messaging
  - Type 9 Packets Support (Data Streaming)
  - Type 11 Message – classification improvements
- DirectIO
  - 8 Load/Store (DirectIO) Units (up from 4)
  - Shadow register sets for LSUs to simplify management and minimize overhead
  - Provide up to 1MB block transfers (up from 4KB)
- Packet Forwarding with Reset Isolation

RapidIO – Topology Examples



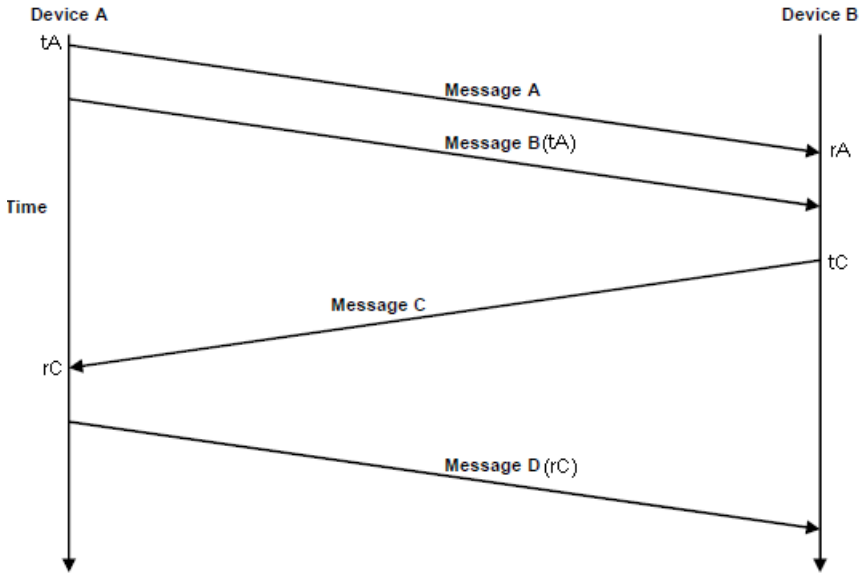




Packet Accelerator Subsystem On Shannon

- 3 Port Ethernet Switch
  - Port 0: Internal hardware queue port
  - Port 1: SGMII 0 Port, 1Gbps
  - Port 2: SGMII 1 Port, 1Gbps
- Packet Accelerator (PA)
  - L2, L3, and L4 packet processing
  - 1.5M packets per sec
- Security Accelerator (SA)
  - Encryption/Decryption
    - IPSEC ESP
    - IPSEC AH
    - SRTP
    - 3GPP

IEEE 1588 support



- Device A is the master device
- Device B is the slave device

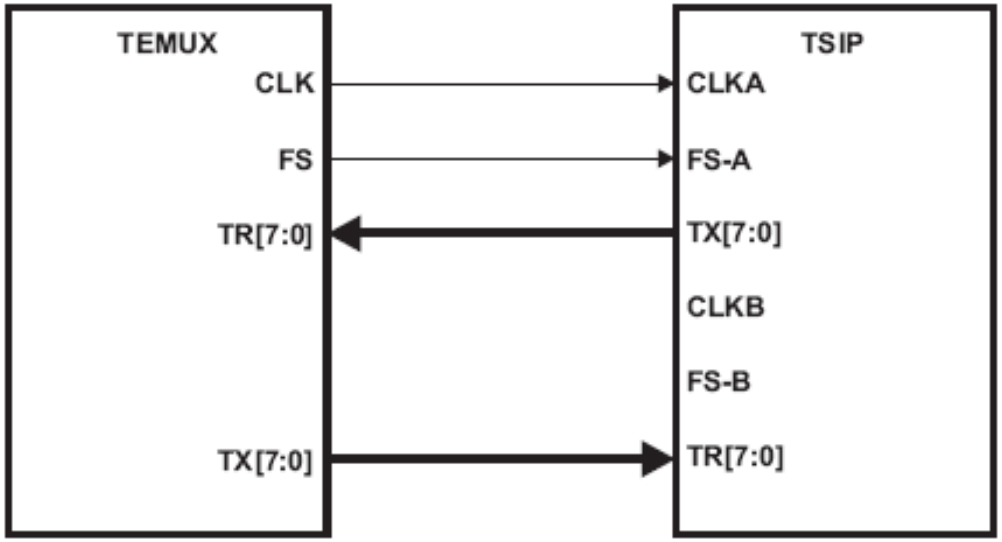
- Message B is used to send the actual transmit time (tA) of Message A
- Message D is used to send the actual receive time (rC) of Message C

- wire time in one direction ((rC - tA)-(tC - rA))/2

- EMAC hardware supports classifying at the physical level ingress and egress frames as timing synchronization frames and the timestamp is recorded.
- A software algorithm running on DSP core would then run the algorithm to calculate the delay and adjust local time accordingly.

TSIP Overview

- 1024 8-bit timeslots receive and transmit.
  - 8 links of 128 timeslots at 8.192 Mbps.
  - 4 links of 256 timeslots at 16.384 Mbps.
  - 2 links of 512 timeslots at 32.768 Mbps.
- Two clock and frame sync inputs.
  - Independent clocking – 1 receive clock and 1 transmit clock.
  - Redundant/common clocking – 1 receive/transmit clock with second clock as backup.



Shannon PCIe Interface

- **Nyquist/Shannon incorporates PCIe interface with the following characteristics:**
  - **Two SERDES lanes running at 5 GBaud/2.5GBaud**
  - **Gen2 compliant**
  - **Three different operational modes (default defined by pin inputs at power up; can be overwritten by software):**
    - **Root Complex (RC)**
    - **End Point (EP)**
    - **Legacy End Point**
  - **Single Virtual Channel (VC)**
  - **Single Traffic Class (TC)**
  - **Maximum Payloads**
    - **Egress – 128 bytes**
    - **Ingress – 256 bytes**
  - **Configurable BAR filtering, IO filtering and configuration filtering**

Remaining Peripherals & System Elements (1/2)

- **EMIF16**
  - Supports NAND flash memory, up to 256MB
  - Supports NOR flash up to 16MB
  - Supports asynchronous SRAM mode, up to 1MB
  - Used for booting, logging, announcement, etc.
- **64-Bit Timers**
  - Total of 16 64-bit timers
    - One 64-bit timer per core is dedicated to serve as a watchdog (or may be used as a general purpose timer)
    - Eight 64-bit timers are shared for general purpose timers
  - Each 64-bit timer can be configured as two individual 32-bit timers
  - Timer Input/Output pins
    - Two timer Input pins
    - Two timer Output pins
    - Timer input pins can be used as GPI
    - Timer output pins can be used as GPO

Remaining Peripherals & System Elements (2/2)

- **UART Interface – Operates at up to 128,000 baud**
- **I2C Interface**
  - **Supports 400Kbps throughput**
  - **Supports full 7-bit address field**
  - **Supports EEPROM size of 4 Mbit**
- **SPI Interface**
  - **Operates at up to 66MHz**
  - **Supports two chip selects**
  - **Support master mode**
- **GPIO Interface**
  - **16 GPIO pins**
  - **Can be configured as interrupt pins**
  - **Interrupt can select either rising edge or falling edge**

Q&A