

Bandwidth Management in DM814x / DM385

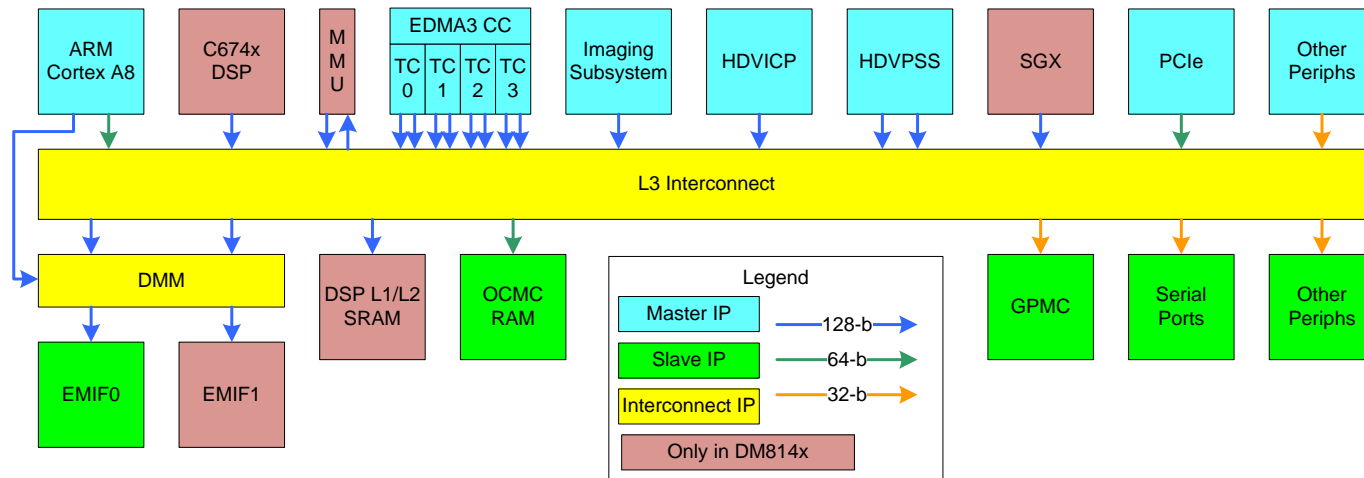
11 June 2013

1

DM385 and DM814x

- DM814x and DM385 Interconnect / DMM are same, except
 - In DM385,
 - C674x DSP is NOT present
 - DSP L1/L2 RAM is NOT present
 - MMU (used by DSP) is NOT present
 - EMIF1 is NOT present
 - SGX is NOT present
 - In DM814x
 - SATA1 is NOT present
- Other differences between DM814x and DM385 are mentioned in the slides where ever applicable

DM814x/DM385 Interconnect overview

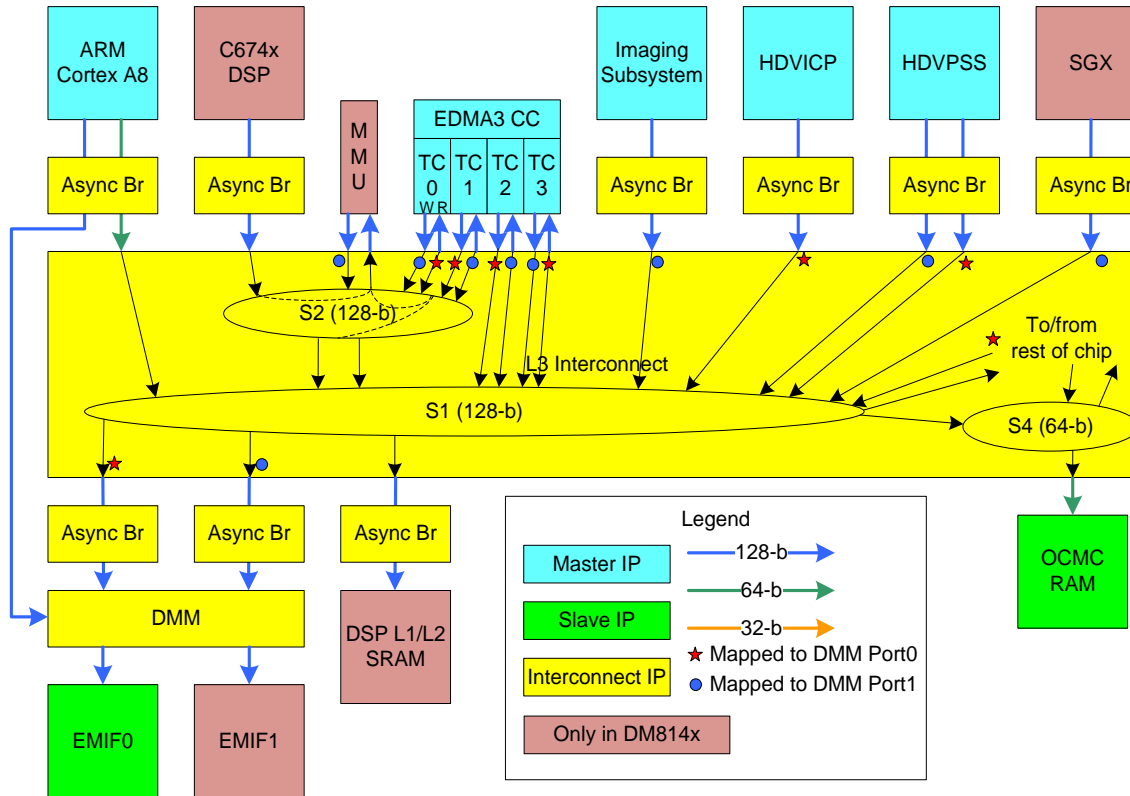


- Master IP – Initiates bus requests
- Slave IP – Responds to bus requests
- L3 Interconnect – Routes/arbitrates bus requests between Masters and Slaves
- Dynamic Memory Manager (DMM)
 - Provides interleaved view of two EMIF's in single address space (DM814x)
 - Provides non-interleaved view of single EMIF in single address space (DM385)
- External Memory Interface (EMIF) - Queues/schedules requests to DRAM

Interconnect Key Characteristics

- Bandwidth
 - Per Interconnect link (128b links)
 - Up to (L3 MHz) $200 \text{ MHz} * 16\text{B/cycle} * 88\% = 2.8 \text{ GBps}$ (*refer to device datasheet for clock rate*)
 - 88% represents peak efficiency due to packet overhead
 - Refer to device datasheet for information on link mapping to L3 clock domain and link width.
 - EMIF/DDR
 - DM814x
 - Up to $400 \text{ MHz} * 2 \text{ (for ddr)} * 4\text{B/ddr edge} * 2 \text{ ports} = 6.4 \text{ GBps}$ (Theoretical) (*refer to device datasheet for clock rate and width*)
 - DM385
 - Up to $400 \text{ MHz} * 2 \text{ (for ddr)} * 4\text{B/ddr edge} * 1 \text{ ports} = 3.2 \text{ GBps}$ (Theoretical) (*refer to device datasheet for clock rate and width*)
 - Practical DDR bandwidth is 50-55% of theoretical DDR BW

DM814x / DM385 Detailed Connectivity for key masters/slaves



- ARM
 - Minimal latency to DDR space by using direct path thru DMM
 - Bypasses interconnect
- DSP
 - Always uses MMU path
 - MMU can be disabled if not needed
- EDMA TC0 and TC1
 - Can optionally use MMU path (in DM814x ONLY), based on MMR setting.
- EDMA TC2 and TC3
 - Routed directly thru S1 to maximize concurrency where required
- DMM Mapping
 - ~1/2 of IP mapped to DMM Port0
 - ~1/2 of IP mapped to DMM Port1
- S2: MMU Loopback switch
- S1: Provides crossbar connectivity between 128-b masters and each memory⁵

Bandwidth Management Overview

- DM814x has Cortex-A8 , HDVICP , HDVPSS , EDMA , Ducati /M3, DSP , USB , GMAC , ISS, etc as data traffic initiators.
- DM385 has Cortex-A8 , HDVICP , HDVPSS , EDMA , Ducati /M3, USB , GMAC , etc as data traffic initiators.
- Above initiators transfer data to/from targets such as DDR memory, OCMC RAM , other processors memory & peripherals.
- Each initiator have programmable
 - pressure control for interconnect.
 - priority control for EMIF
- This would enable each initiator to get latency and/or bandwidth they require.

L3 Interconnect Pressure

- Pressure controlled independently for each initiator.
- 3 pressure levels
 - 0 = Lowest, 1 = Middle, 3 = highest
 - round robin arbitration within a given pressure level.
- Determines which pending bus requests to a given slave wins arbitration in a switch
 - E.g., controls which concurrent request is sent to EMIF/DMM next
- ISS
 - **BW regulator** dynamically controls pressure
 - No Pressure bits to control priority statically
- HDVPSS
 - **Bit0 IP Controlled Dynamic**
 - Custom scheme based on internal FIFO status
 - Based on margin to overflow/underflow
 - **Bit1 MMR Controlled Static (INIT_PRIORITY_n)**
- PCIe, USB, EMAC, EDMA_TC0, TC2:
 - **Statically programmed**
 - Via chip level MMR (INIT_PRIORITY_n).
- C674x DSP (via MMU) *, EDMA_TC1, TC3, HDVICP, SGX * :
 - **BW regulator** dynamically controls pressure
 - * ONLY in DM814x

MMR based Pressure settings

- Registers to set L3 Pressure via INIT_PRIORITY_0 & INIT_PRIORITY_1 in control module.

INIT_PRIORITY_0 : 0x48140608

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
[Black]				TCWR2		TCRD2		[Black]				TCWR0		TCRD0		[Black]				HDVPSS1		[Black]	HDVPSS0	System MMU		GEM_CFG	GEM_MDMA	HOST_ARM			

Only in DM814x

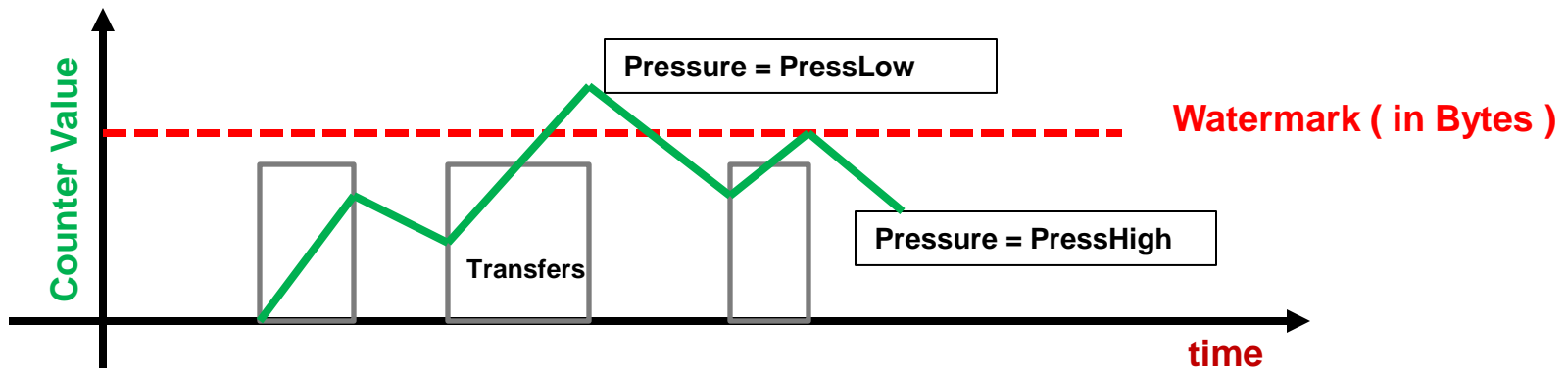
INIT_PRIORITY_1 : 0x4814060C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
[Black]										SGX		[Black]		PCIE		M3Ducati		[Black]		SATA1	SATA	USB_QMGR		USB_DMA	[Black]		CPGMAC0				

Only in DM385

Bandwidth Regulator

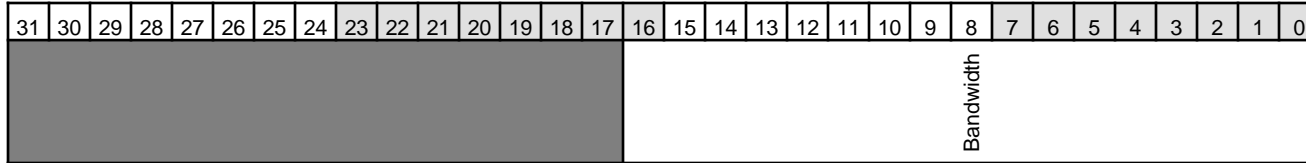
- For a given initiator:
- Increases pressure when the actual consumed bandwidth is lower than expected bandwidth
- Decreases pressure once the expected bandwidth is reached.
- Mechanism
 - A counter is incremented by number of bytes transferred (read + write)
 - At each clock cycle, a quantity corresponding to expected bandwidth is subtracted from the counter.
 - A Watermark value for the counter is programmed.
 - When counter value is less than Watermark high pressure (as define by PressHigh) is applied,
 - Else low pressure (as defined by PressLow) is applied.



* Traffic pattern is for illustration only

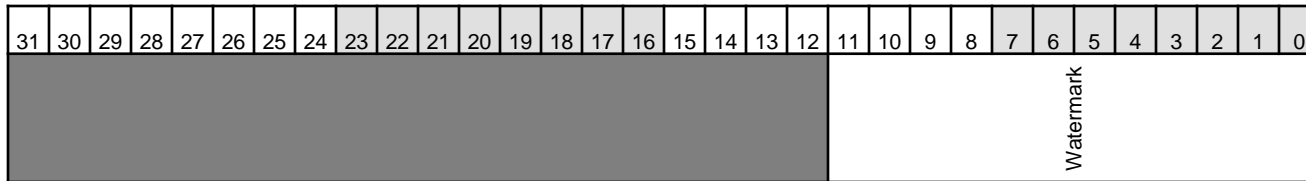
Setting up a Bandwidth Regulator

Bandwidth : 0x08



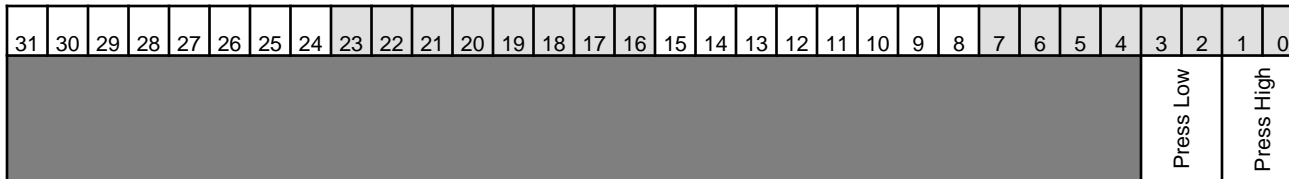
Required Bandwidth
Bus Freq / (2⁵)

Watermark: 0x0C



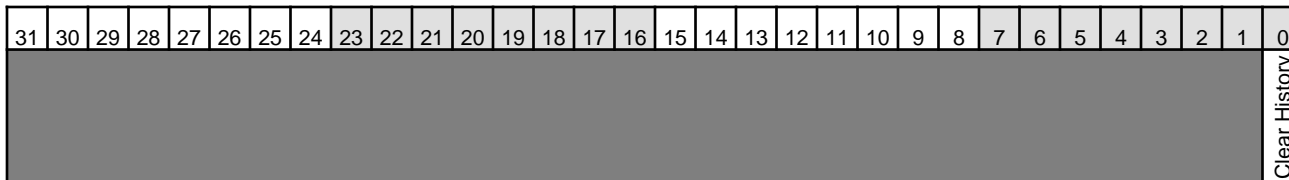
MovingWindow * Bandwidth

Press: 0x10



Press Low should be less than equal to Press High

Clear History : 0x014



Write 1 after updating other registers

Bandwidth Regulator Base Address

Bandwidth Regulator name	Base Address
HDVICP0_BW_REGULATOR	0x44401C00
EDMA_RD3_BW_REGULATOR	0x44001F00
EDMA_WR3_BW_REGULATOR	0x44002000
EDMA_RD1_BW_REGULATOR	0x44002100
EDMA_WR1_BW_REGULATOR	0x44002200
MMU_BW_REGULATOR (DSP)	0x44002300
SGX_BW_REGULATOR	0x44402400
ISS_BW_REGULATOR	0x44402500

Only in DM814x

Example for DSP Bandwidth Regulator programming

- **Intent** - DSP should have minimal latency but should not take excessive bandwidth
- **Details**
 - L3 Interconnect = 200 MHz
 - Highest Pressure for DSP accesses by default (for low latency)
 - Low Pressure if Bandwidth exceeds 100 MB/s
 - Compute watermark over a 200 interconnect cycle interval, or 1us
- **Calculation**
 - Bandwidth register => $100\text{MBps} / (200\text{ MHz}/2^5) = 16 = 0x10$
 - Watermark register => $1\text{ us} * 100\text{ MBps} = 100 = 0x64$
 - Pressure Register => { PressLow = 0x0, PressHigh = 0x3 }
 - Start Bandwidth Reg by writing 0x1 to Clear History register

Example for HDVICP Bandwidth Regulator programming

- **Intent** - HDVICP should have 1GB/s bandwidth & should not take excessive bandwidth
- **Method**
 - L3 Interconnect = 200 MHz
 - Medium Pressure for HDVICP accesses by default (to ensure bandwidth)
 - Low Pressure if Bandwidth exceeds 1GBps
 - Compute watermark over 500 interconnect cycles, or 2.5 us
- **Calculation**
 - Bandwidth register => $1000\text{MB/s} / (200\text{ MHz}/2^5) = 160 = 0xA0$
 - Watermark register => $2.5\text{ us} * 1000\text{ MB/s} = 2500 = 0x9C4$
 - Pressure Register => { PresLow = 0x0 ,PressHigh = 0x1 }
 - Start Bandwidth Reg by writing 0x1 to Clear History register

Priority Control in EMIF

- Every initiator except HDVPSS there is a priority configuration in DMM PEG registers
- HDVPSS priority is programmed in VPDMA descriptor
- Priority is 3 bit field (0 ... 7) , 0 is highest priority, 7 is lowest
- Priority determines prioritization of data transfers in EMIF

Configuring DMM PEG

DMM_PEG_PRI07 : 0x63C															
31	30...28	27	26...24	23	22...20	19	18...16	15	14...12	11	10...8	7	6...4	3	2...0
PRIO ₆₃		PRIO ₆₂		PRIO ₆₁		PRIO ₆₀		PRIO ₅₉		PRIO ₅₈		PRIO ₅₇		PRIO ₅₆	
W7	P7	W6	P6	W5	P5	W4	P4	W3	P3	W2	P2	W1	P1	W0	P0

.....

DMM_PEG_PRI00 : 0x620															
31	30...28	27	26...24	23	22...20	19	18...16	15	14...12	11	10...8	7	6...4	3	2...0
PRIO ₇		PRIO ₆		PRIO ₅		PRIO ₄		PRIO ₃		PRIO ₂		PRIO ₁		PRIO ₀	
W7	P7	W6	P6	W5	P5	W4	P4	W3	P3	W2	P2	W1	P1	W0	P0

the 3-bit priority coded on the 3 least significant bits (0 is the higher priority)

A “W” field-specific active-high local write enable bit, always read as 0

The role of the W bit is to allow the modification of a single entry without requiring a read-modify-write sequence.

DMM PEG Registers

Initiator	Register	Register Address	Priority Field
CortexA8	DMM_PEG_PRI00	0x4E00_0620	PRI00
System MMU	DMM_PEG_PRI01	0x4E00_0624	PRI010
Ducati	DMM_PEG_PRI01	0x4E00_0624	PRI014
SATA1	DMM_PEG_PRI02	0x4E00_0628	PRI016
TPTC0 Read	DMM_PEG_PRI03	0x4E00_062C	PRI024
TPTC1 Read	DMM_PEG_PRI03	0x4E00_062C	PRI025
TPTC2 Read	DMM_PEG_PRI03	0x4E00_062C	PRI026
TPTC3 Read	DMM_PEG_PRI03	0x4E00_062C	PRI027
TPTC0 Write	DMM_PEG_PRI03	0x4E00_062C	PRI028
TPTC1 Write	DMM_PEG_PRI03	0x4E00_062C	PRI029
TPTC2 Write	DMM_PEG_PRI03	0x4E00_062C	PRI030
TPTC3 Write	DMM_PEG_PRI03	0x4E00_062C	PRI031
SGX530	DMM_PEG_PRI04	0x4E00_0630	PRI032
HDVICP0	DMM_PEG_PRI05	0x4E00_0634	PRI040
ISS	DMM_PEG_PRI05	0x4E00_0634	PRI044
GMAC0	DMM_PEG_PRI06	0x4E00_0638	PRI048
USB DMA	DMM_PEG_PRI06	0x4E00_0638	PRI052
USB QMGR	DMM_PEG_PRI06	0x4E00_0638	PRI053
SATA0	DMM_PEG_PRI07	0x4E00_063C	PRI057
PCIe	DMM_PEG_PRI07	0x4E00_063C	PRI058

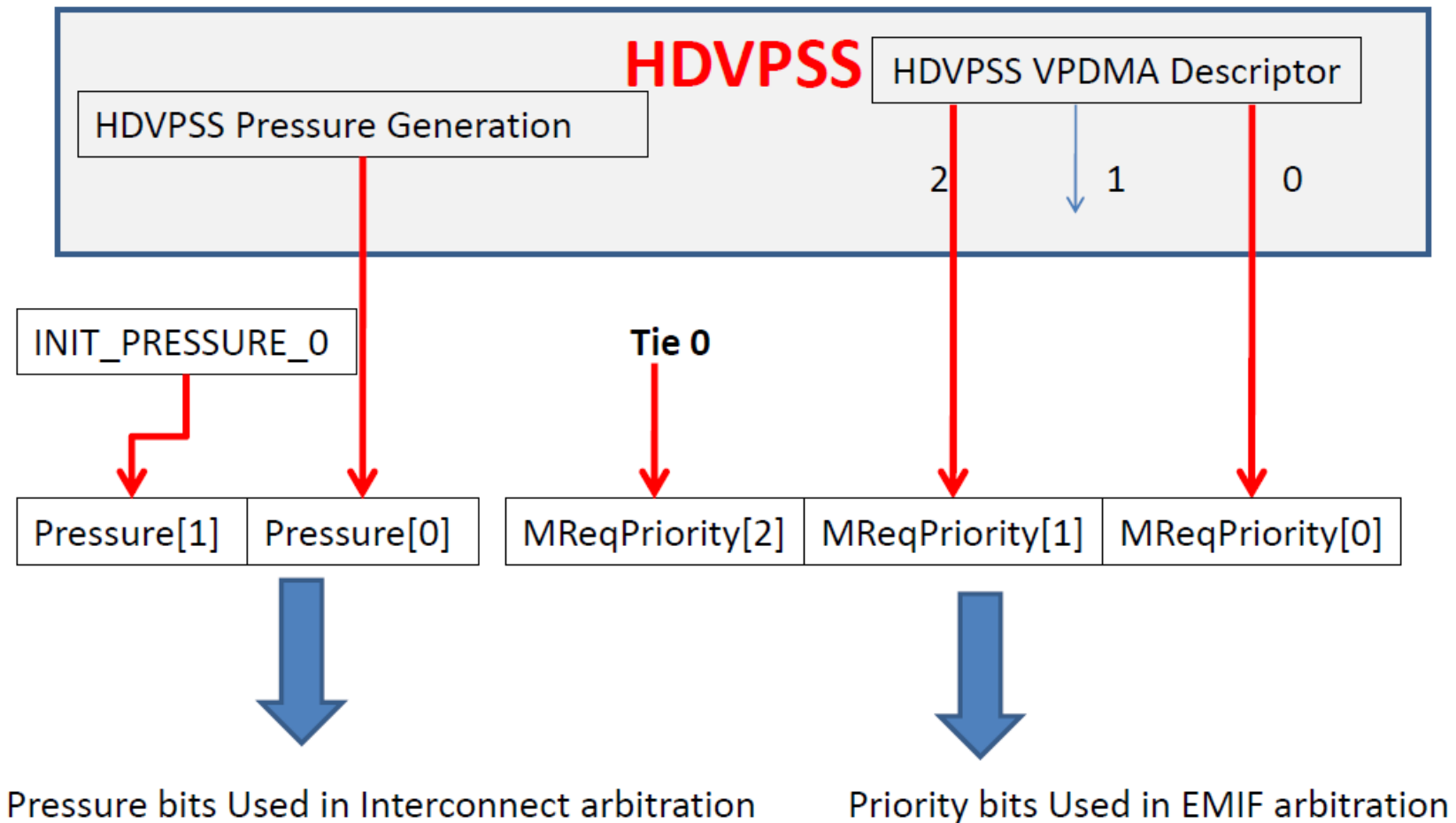
Only in DM814x

Only in DM385

EMIF Priority setting through DMM example

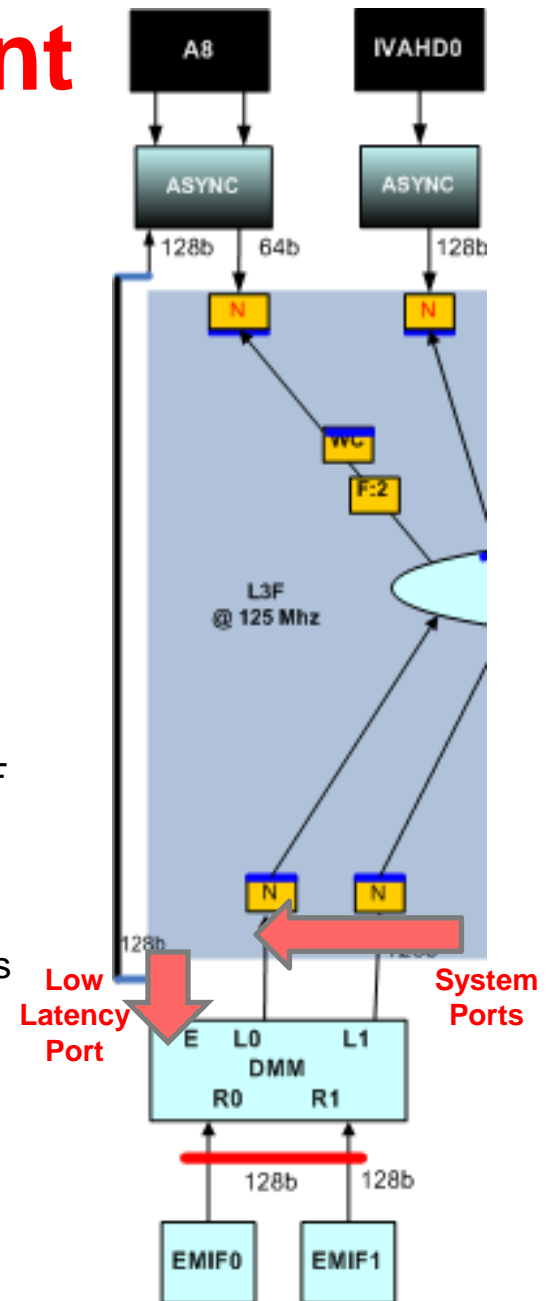
- Set Ducati/M3 Priority of 0x1
 - Register DMM_PEG_PRI01 , Field PRIO14 (Bits 27-24) would be used to change ducati priority
 - DMM_PEG_PRI01 address = 0x4E00_0624
 - Data to be written (0b1001) << 24 = 0x0900_0000
 - Once Data is written , Field PRIO14 (Bits 27-24) would reflect value as 0b0001
- Note: DMM_PEG_PRI0x registers doesn't need read-modify-write sequence

HDVPSS pressure & priority settings

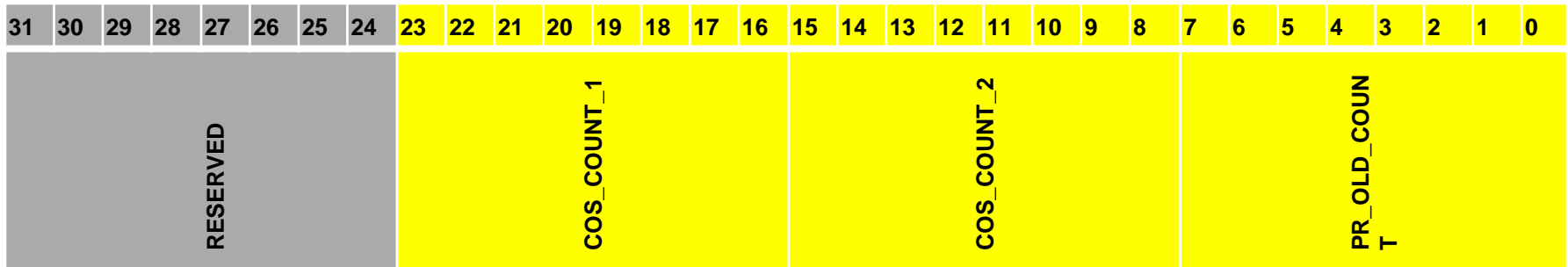


A8 Priority Management

- DM81xx L3 architecture provides DDR access to the system via two paths
 - Low latency port to ARM (A8)
 - System access ports (Rest of peripherals)
- In order to implement better priority arbitration between A8 and rest of the peripherals, its important to program the following registers to enable class of service .
 - PBBPR register
 - [23:16]COS_COUNT_1 : Priority Raise Counter for class of service 1. Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 1 commands in the Command FIFO. A value of N will be equal to $N \times 16$ clocks.
 - [15:8]COS_COUNT_2 : Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 2 commands in the Command FIFO. A value of N will be equal to $N \times 16$ clocks.
 - [7:0]PR_OLD_COUNT : Number of memory transfers after which the EMIF momentarily raises the priority of old commands in the OCP Command FIFO.
 - DMM Priority



Configuring PBBPR

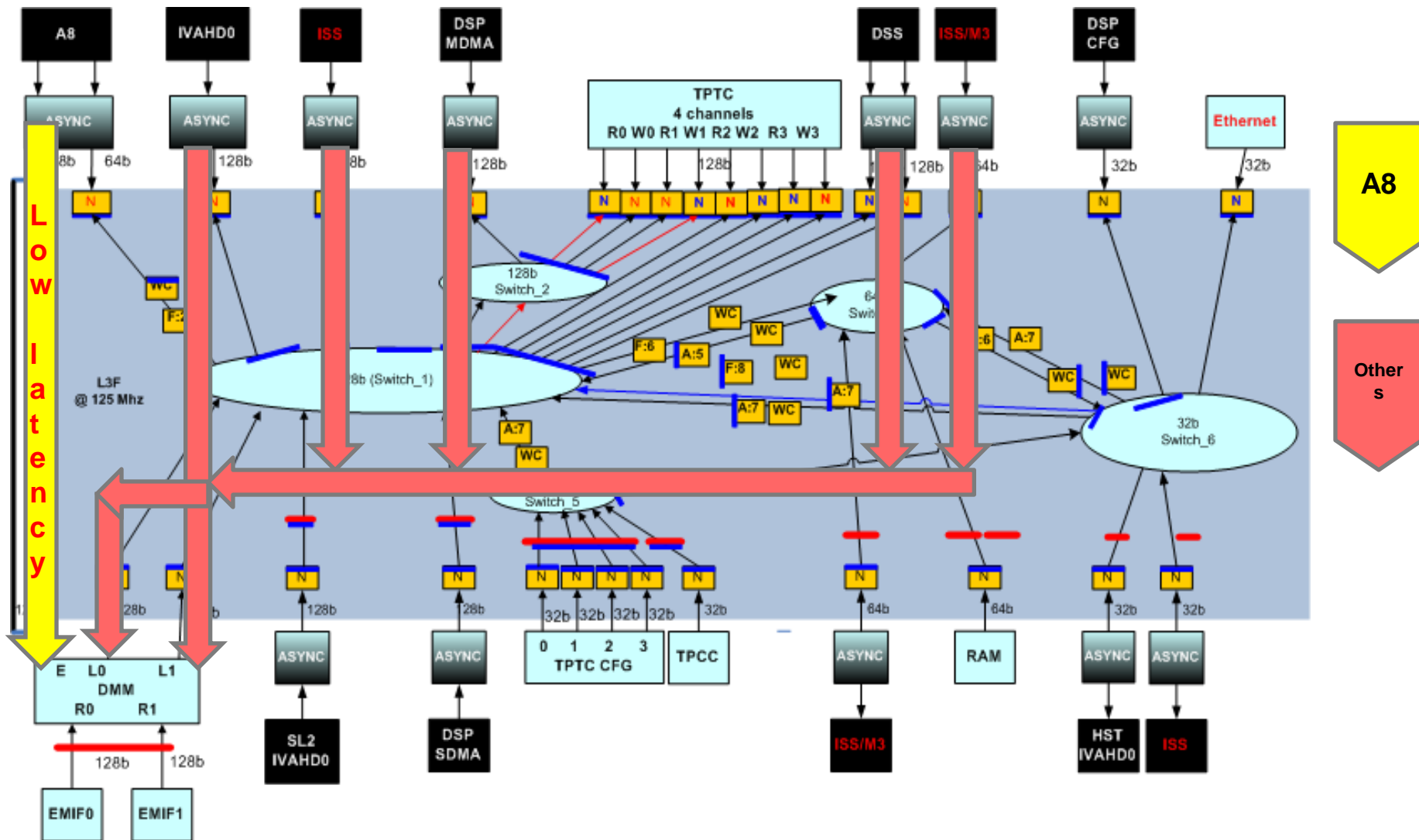


- **PBBPR: (EMIF4_0_CFG_BASE + 0x54), (EMIF4_1_CFG_BASE + 0x54)***
 - **[23:16] COS_COUNT_1**
 - Priority Raise Counter for class of service 1. Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 1 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.
 - MAX = 0xFF
 - MIN = 0x0 (defaults to 1)
 - **Recommended : Lower than default (needs system testing)**
 - **[15:8] COS_COUNT_2**
 - Number of m_clk cycles after which the EMIF momentarily raises the priority of the class of service 2 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.
 - MAX = 0xFF
 - MIN = 0x0 (defaults to 1)
 - **Recommended : DEFAULT**
 - **[7:0] PR_OLD_COUNT**
 - Number of memory transfers after which the EMIF momentarily raises the priority of old commands in the OCP Command FIFO.
 - MAX = 0xFF
 - MIN = 0x0 (defaults to 1)
 - **Recommended : 0x10 – 0x60 (needs system test)**

*Not valid for DM385

ISS Based applications - Priority Management

BW competitors for ISS

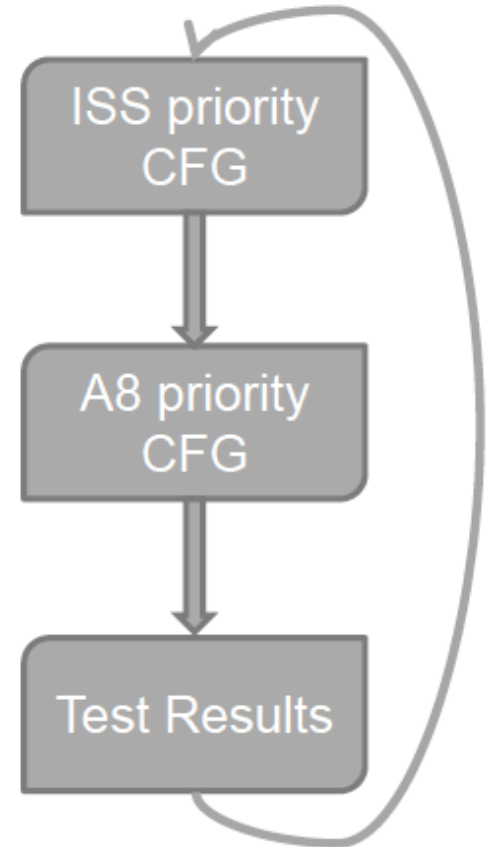


ISS priority control

- Following should be following in the given order:
 - ISS_BW_REGULATOR
 - This should be the first knob to step up the ISS priority.
 - Set PRESS_LOW and PRESS_HIGH to either '2' or '3' to setup static level 2 or level 3 pressure on ISS to DDR path.
 - DMM PRIORITY
 - Configure DMM PEG priority to make ISS initiator as higher priority (0 is highest) and other initiators (A8, IVA..etc) lower priority.
 - ISS CLKDIV CONTROLS
 - Gradually decrease IPIPEIF_CLKDIV to lowest value which can meet the usecase.
 - Gradually decrease RSZ_CLKDIV from default value of 0xFFFF to reduce RSZ operation speed and thus RSZ DMA out rate. This should help RSZ OVF issues.

How to solve OVF issues?

- Overflows in ISS are a result of insufficient availability of peak bandwidth to ISS DMA. As a result it could result in RSZ, ISIF overflows or IPIPEIF read under-run issues and cause performance losses.
- Tuning system for maximizing ISS bandwidth is typically a 2 step process – first resolve peripheral priority to give ISS top priority and second enable QOS on A8 so that it doesn't deplete DMM/DDR resource.
- **Peripherals priority conflicts**
 - This covers priority arbitration conflicts between peripherals such as ISS and other peripherals such as DSS, IVAHD, DSP..etc
 - To configure ISS priority in such cases, following two priority schemes should be enough:
 - ISS BW REGULATOR
 - Configure ISS BW regulator to prioritize ISS to DDR path with a priority override of '2' or '3' (level). This is similar to setting the L3_PRIO statically with the similar level.
 - DMM PRIORITY
- **ARM vs ISS priority conflicts**
 - In this scenario the conflict is between ISS, DSS..etc and ARM (A8) for DDR priority arbitration. Since A8 has a low latency path to DDR regular DMM_PRIORITY configuration scheme doesn't work well. To configure ISS priority in such cases please follow:
 - BURST PRIO (PBBPR register)
 - Configure COS_COUNT_1, COS_COUNT_2 and PR_OLD_COUNT
 - DMM PRIORITY



Thank You