KeyStone Training

Serial RapidIO (SRIO) Subsystem

Agenda

- SRIO Overview
- DirectIO Operation
- Message Passing Operation
- Other RapidIO Features
- Summary

SRIO Overview

- SRIO Overview
- DirectIO Operation
- Message Passing Operation
- Other RapidIO Features
- Summary

Introduction To RapidIO

- Two Basic Modes of Operation:
  - DirectIO
    - Read/write operations directed to specific memory address
      - Transmit device has knowledge of memory map of receiving device
    - Functional units:
      - LSU (Load/Store Unit)
      - MAU (Memory Access Unit)
  - Message Passing
    - Mailbox and Letter designators
      - Transmit device does not need knowledge of memory map of receiving device
    - Functional units:
      - TXU (Message Transmit Unit)
      - RXU (Message Receive Unit)
- Data communication on differential input/output ports
- Overall RapidIO architecture divided into three layers:
  1. Physical Layer
     - SERDES
     - RapidIO Physical layer IP
  2. Transport Layer
     - Transports packet from physical layer to logical layer protocol units
  3. Logical Layer
     - Protocol Units (e.g. LSUs, TXU, etc.)

New Features Compared to C64x+

- RapidIO 2.1.1 Compliant / RapidIO 1.2 Compliant

Physical Layer

- For ports with options show in the table below:

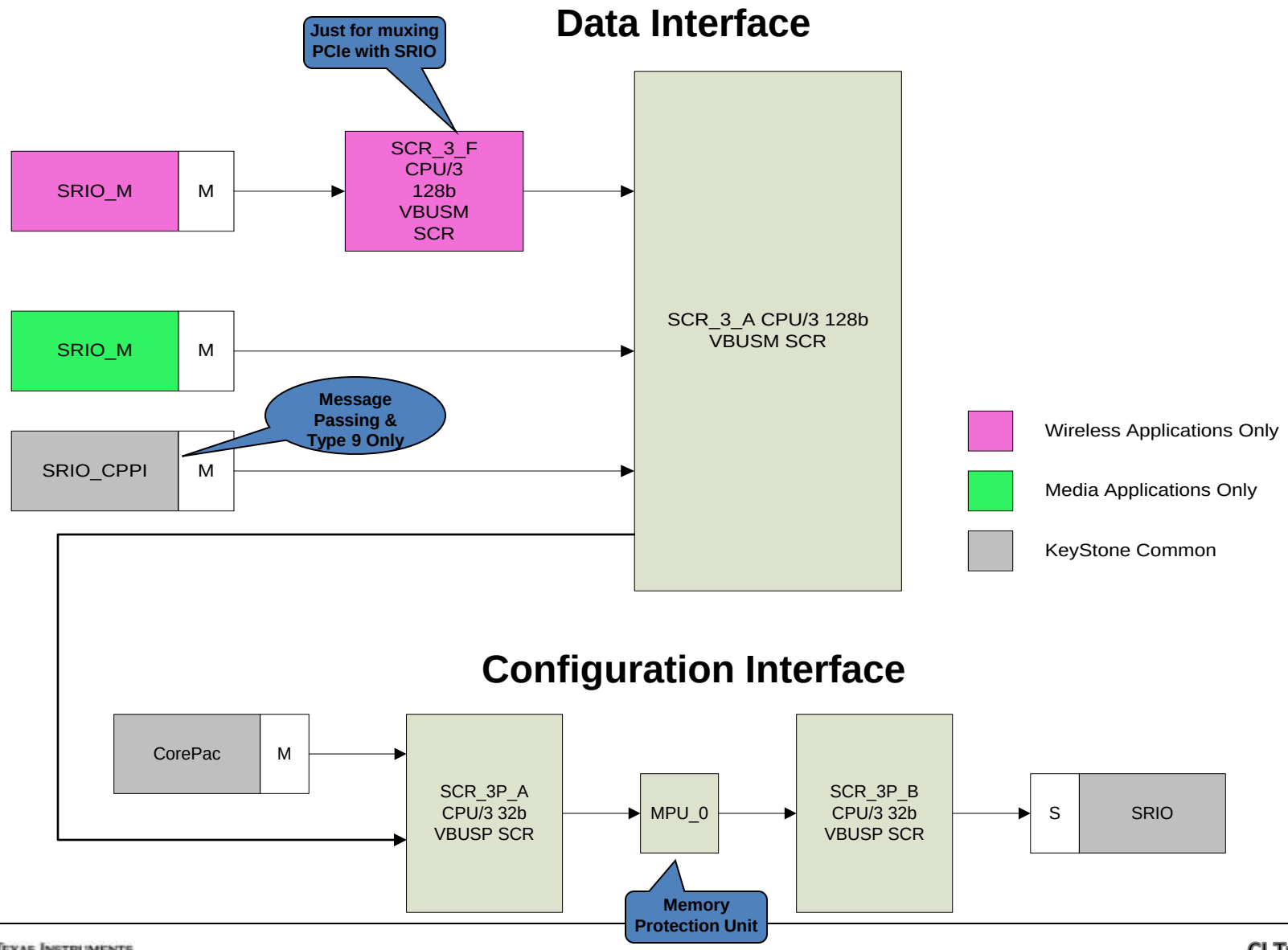
Mode 0	1x	1x	1x	1x
Mode 1	1x	1x	2x	
Mode 2	2x		1x	1x
Mode 3	2x		2x	
Mode 4	4x			

- Data rates up to 5 Gbaud/ Data rates up to 3.125 Gbaud
- Different ports at different baud rates (Only integer multiple different rates are allowed)

Logical Layer

- 24 Interrupt outputs / 8 Interrupt outputs
- 1 MB LSU transaction size with queuing capability/ single 4 KB LSU transaction size
- Type 9 Packet Support (Data Streaming)
- External Type 9 and Type 11 queue management
- Strict priority scheduler / Round-robin scheduler
  - Round robin interleaved on a packet basis at a given priority
  - Outbound credit-aware functional blocks
- 16 Local DeviceIDs & 8 Multicast IDs / 1 Local DeviceID & 3 Multicast IDs
- Auto-promotion of response priorities by RXU and MAU can now be disabled
- Ability to set the CRF (Critical Request Flow) bit on outgoing requests and responses

SRIO in KeyStone Devices



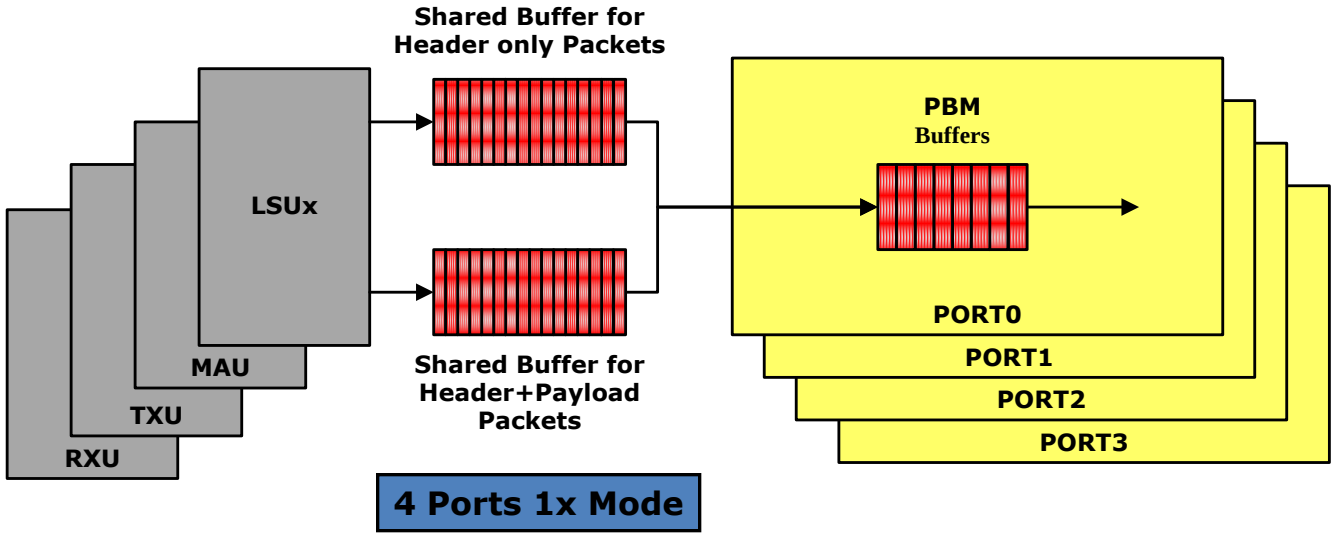
Packet Types

FType	TType (4'b)	Operation
0	don't care	None
1	don't care	None
2	100	NREAD
	1100	Atomic increment
	1101	Atomic decrement
	1110	Atomic set
	1111	Atomic clear
	Others	
3	don't care	None
4	don't care	None
5	4'b0100	NWRITE
	4'b0101	NWRITE_R
	4'b1110	Atomic test & swap
	Others	
6	don't care	SWRITE
7	don't care	Congestion control
8	4'b0000	Maintenance Read
	4'b0001	Maintenance Write
	4'b0010	Maintenance Read Response
	4'b0011	Maintenance Write Response
	4'b0100	Maintenance Port-write
	Others	
9	don't care	Data Streaming
10	don't care	Doorbell
11	don't care	Message
12	don't care	None
13	4'b0000	Response (+Doorbell Response)
	4'b0001	Message Response
	4'b1000	Response w/payload
	other	
14	don't care	None
15	don't care	None

New Packet Type Supported in KeyStone

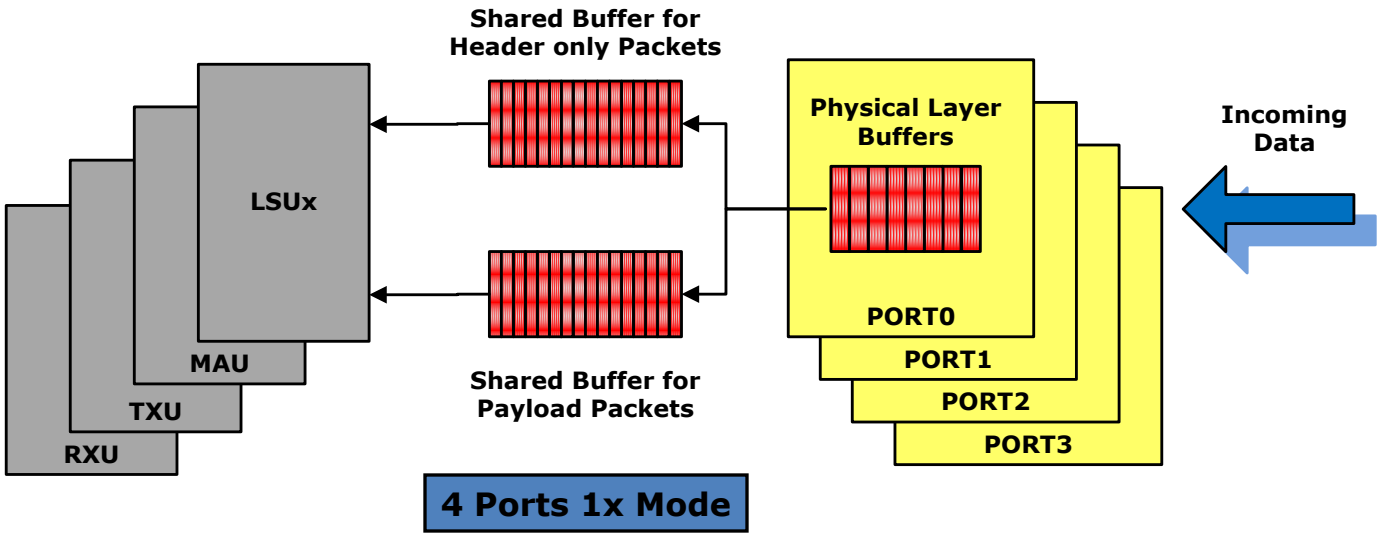
Physical & Logical Layer Considerations

- Tx Buffers
  - 16-deep shared buffer for Tx header and 16-deep shared buffer for Tx header + payload packets
  - If SRIO is configured for four ports in 1x mode, then 8-deep Tx physical layer buffers are supported.
  - If SRIO is configured for two ports in 2x mode, then 16-deep Tx physical layer buffers are supported.
  - If SRIO is configured for one port in 4x mode, then 32-deep Tx physical layer buffers are supported.
- Tx Operation
  - Getting outbound credit:
    - Based on programmed Tx watermarks
    - If the watermarks for a packet indicate that minimum "required buffer count" is three and if SRIO is configured for four ports in 1x mode (8-deep physical layer), then after five buffers are filled, no credit is granted to that PRI of packets.
    - If SRIO is configured for one port in 4x mode (32-deep physical layer), then after 29 buffers are filled, no credit is granted to that PRI of packets.
  - Strict priority-based scheduling:
    - As illustrated by the example shown below, three protocol units -- LSU0, MAU, and TXU -- have packets to send. The priority of those packets are 2, 1 and 2 respectively. The scheduler will round robin between LSU0 and TXU to send data. Because LSU0 and TXU have a higher priority, the transmission of their data must be completed before the scheduler begins to send data from MAU.
    - Protocol units know whether they have outbound credit or not. If a protocol unit does not have outbound credit, then it will not be part of round robin scheduling.
  - Data reaches from shared buffer memory to physical layer memory and goes out in the same order unless there is physical layer re-ordering due to retry or any other condition.

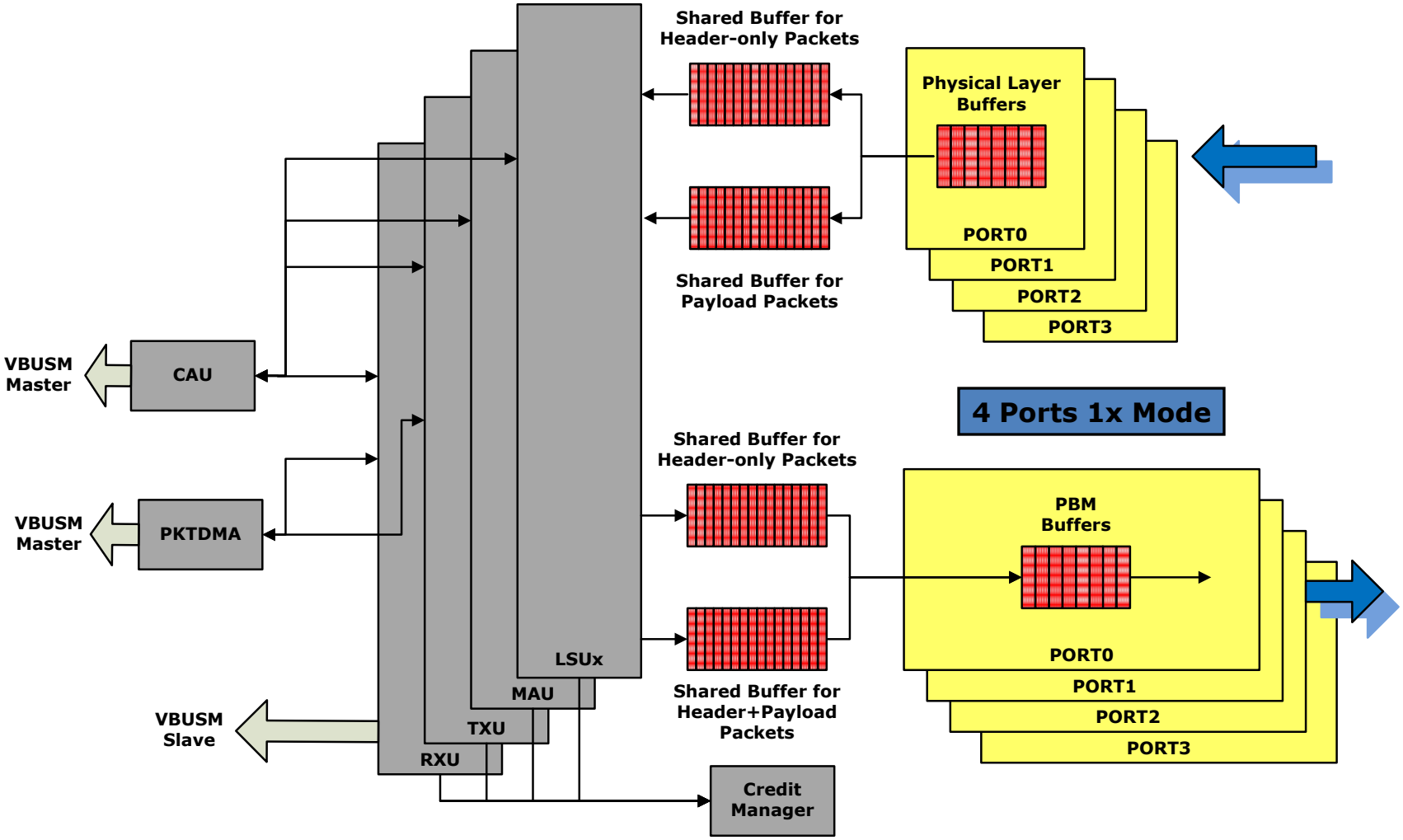


Physical & Logical Layer Considerations

- Rx Buffers
  - 16-deep shared buffer for Rx header and 16-deep shared buffer for Rx header+payload packets
  - If SRIO is configured for four ports in 1x mode, then 8-deep physical layer Rx buffers are supported.
  - If SRIO is configured for two ports in 2x mode, then 16-deep physical layer Rx buffers are supported.
  - If SRIO is configured for one port in 4x mode, then 32 deep physical layer Rx buffers are supported.
- Rx Operation
  - Incoming data reaches to physical Layer
  - Round Robin pickup from four ports to move to shared buffer
  - No priorities are considered while moving from the physical layer buffers to shared buffer



Functional Diagram



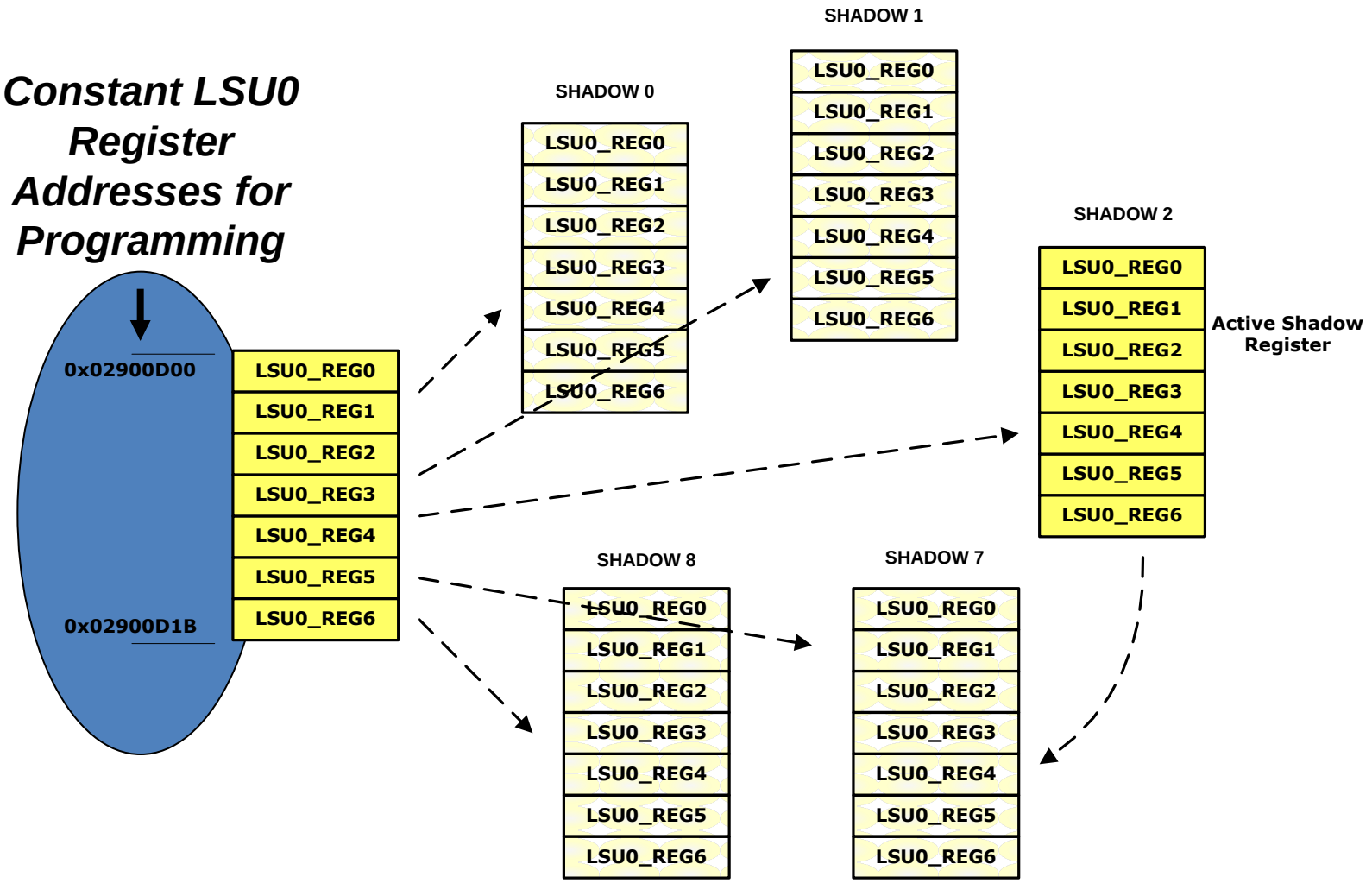
DirectIO Operation

- SRIO Overview
- DirectIO Operation
- Message Passing Operation
- Other RapidIO Features
- Summary

C66x DirectIO Operations Compared to C64x+

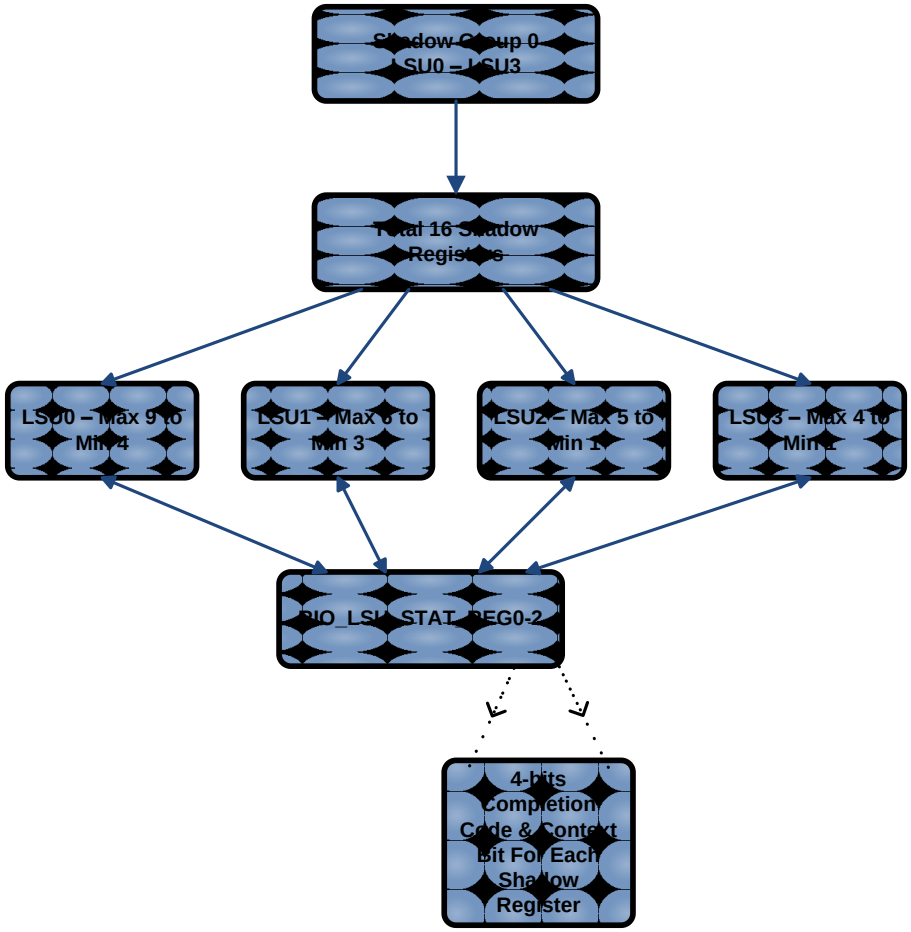
- 8 LSUs / 4 LSUs
- Maximum transaction size (byte_count field) of 1MB / 4KB
  - Up to 4K packets of 256 bytes per LSU programming
- Shadow Registers Concept
- 128 outstanding non-posted packets in total, 16 per LSU (not configurable)
- Auto-generation of doorbell at the end of transfer completion.
  - Send doorbell after sending last packet.
    OR
  - Send doorbell after receiving last response.
  - No doorbell is sent if there is an error.
- Restart and flush LSU transactions.

Shadow Registers Example



Shadow Register Combinations

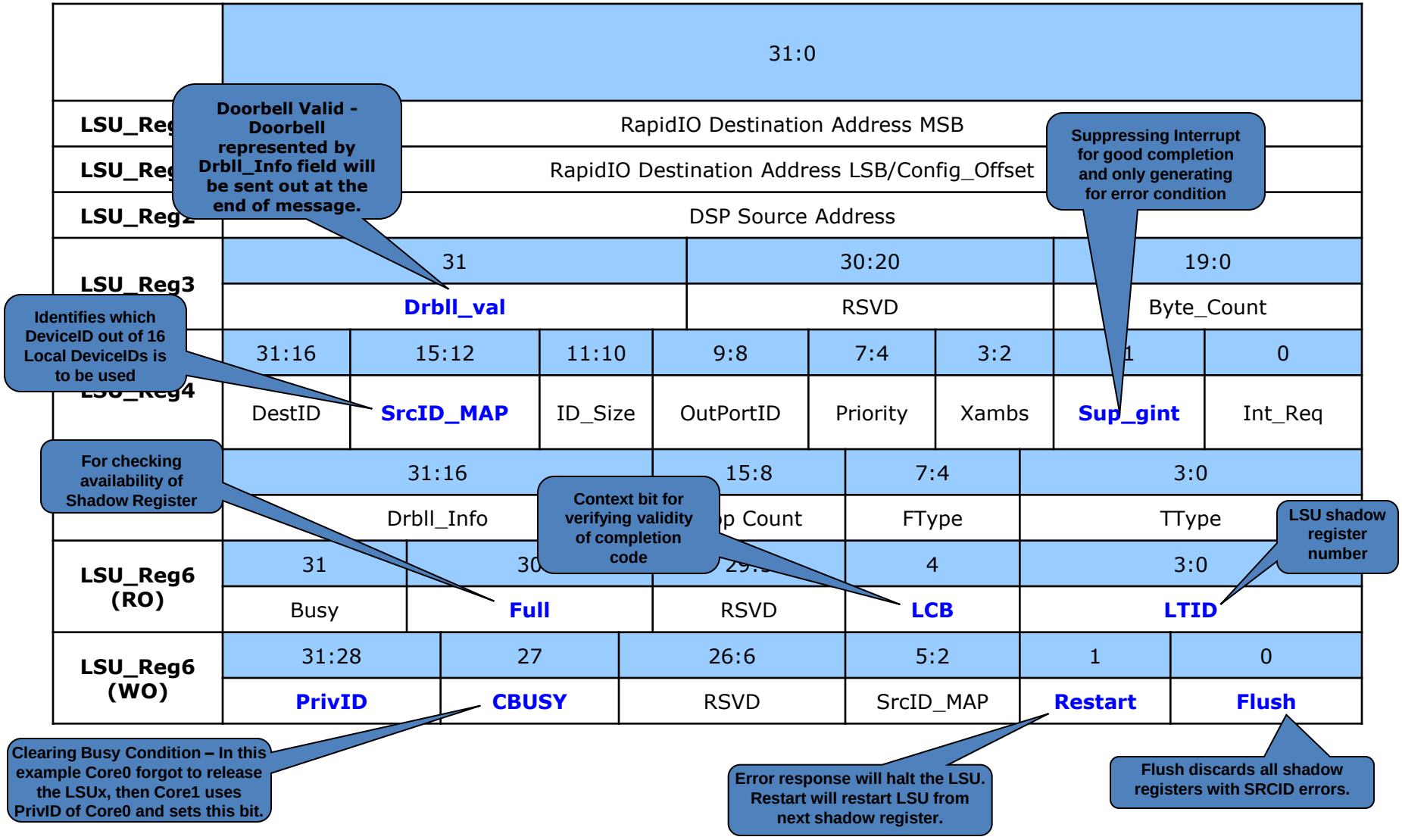
- Same LSU registers, so no memory map address change.
- Two Shadow Register groups:
  - Shadow Group 0 for LSU0 to LSU3
  - Shadow Group 1 for LSU4 to LSU7
- Total of 32 shadow registers with a maximum of 16 per group
- Shadow Register group restrictions:
  - Pre-defined combinations of shadow registers per LSU. The diagram shown here identifies those combinations.
  - Each LSU will have at least one shadow register.
  - Maximum of nine shadow registers per LSU
- RIO_LSU_SETUP_REG0 register used for storing this number for each LSU.
- RIO_LSU_STAT_REG0 set to 2 stores completion code for each shadow register set of Shadow Group0.
- Same mechanism applies to Shadow Group1 for LSU3 to LSU7.



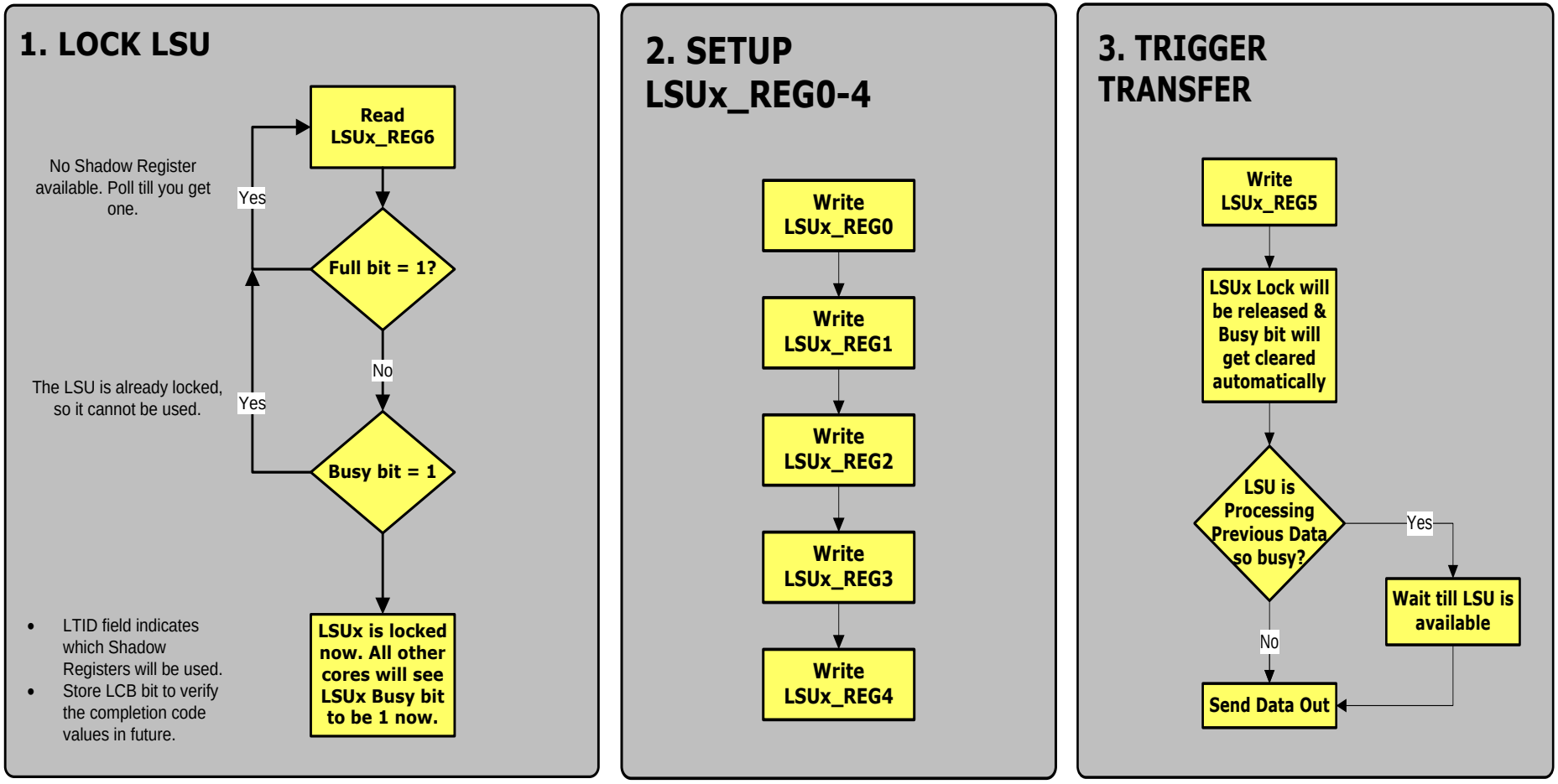
Shadow Register Pre-defined Combinations

Configuration #	LSU0/LSU4	LSU1/LSU5	LSU2/LSU6	LSU3/LSU7
5'h00	4	4	4	4
5'h01	5	5	5	1
5'h02	5	5	4	2
5'h03	5	5	3	3
5'h04	5	4	4	3
5'h05	6	6	3	1
5'h06	6	6	2	2
5'h07	6	5	4	1
5'h08	6	5	3	2
5'h09	6	4	4	2
5'h0A	6	4	3	3
5'h0B	7	6	2	1
5'h0C	7	5	3	1
5'h0D	7	5	2	2
5'h0E	7	4	4	1
5'h0F	7	4	3	2
5'h 10	7	3	3	3
5'h 11	8	6	1	1
5'h 12	8	5	2	1
5'h 13	8	4	3	1
5'h 14	8	4	2	2
5'h 15	8	3	3	2
5'h 16	9	5	1	1
5'h 17	9	4	2	1
5'h 18	9	3	3	1
5'h 19	9	3	2	2

LSU Registers



Tx Operation: Non-EDMA Mode

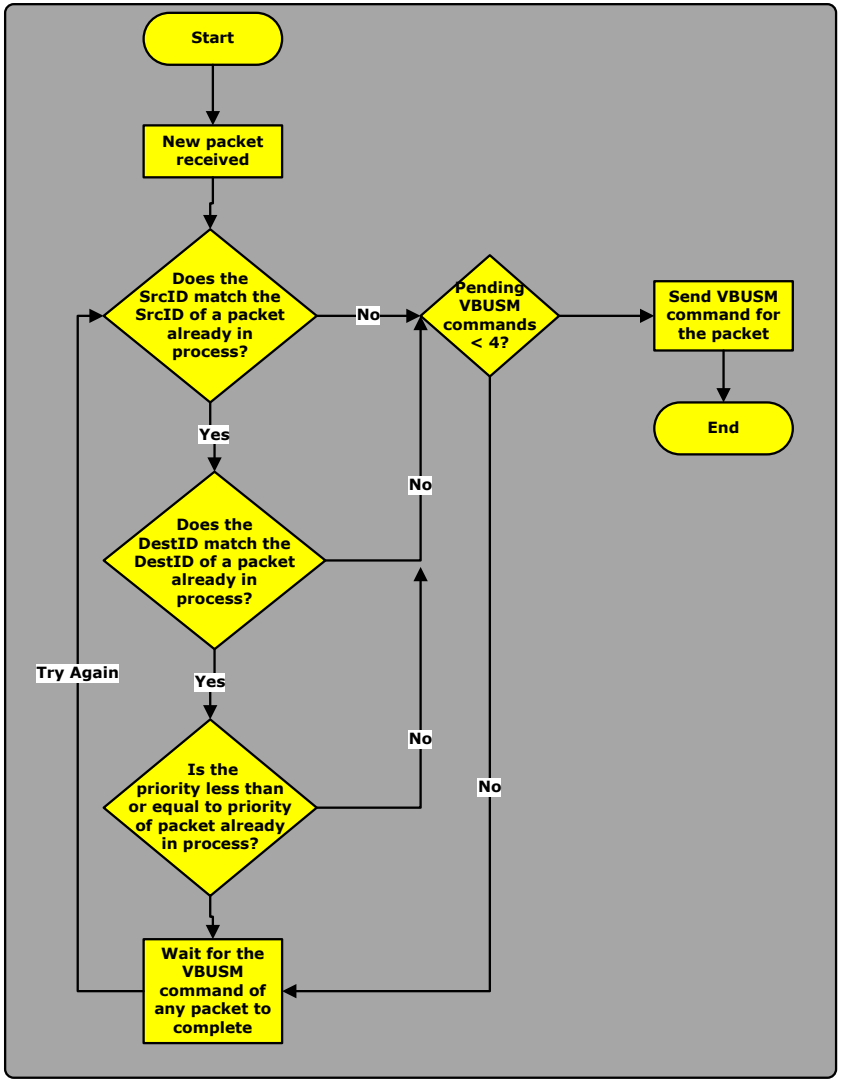


Tx Operation: EDMA Mode

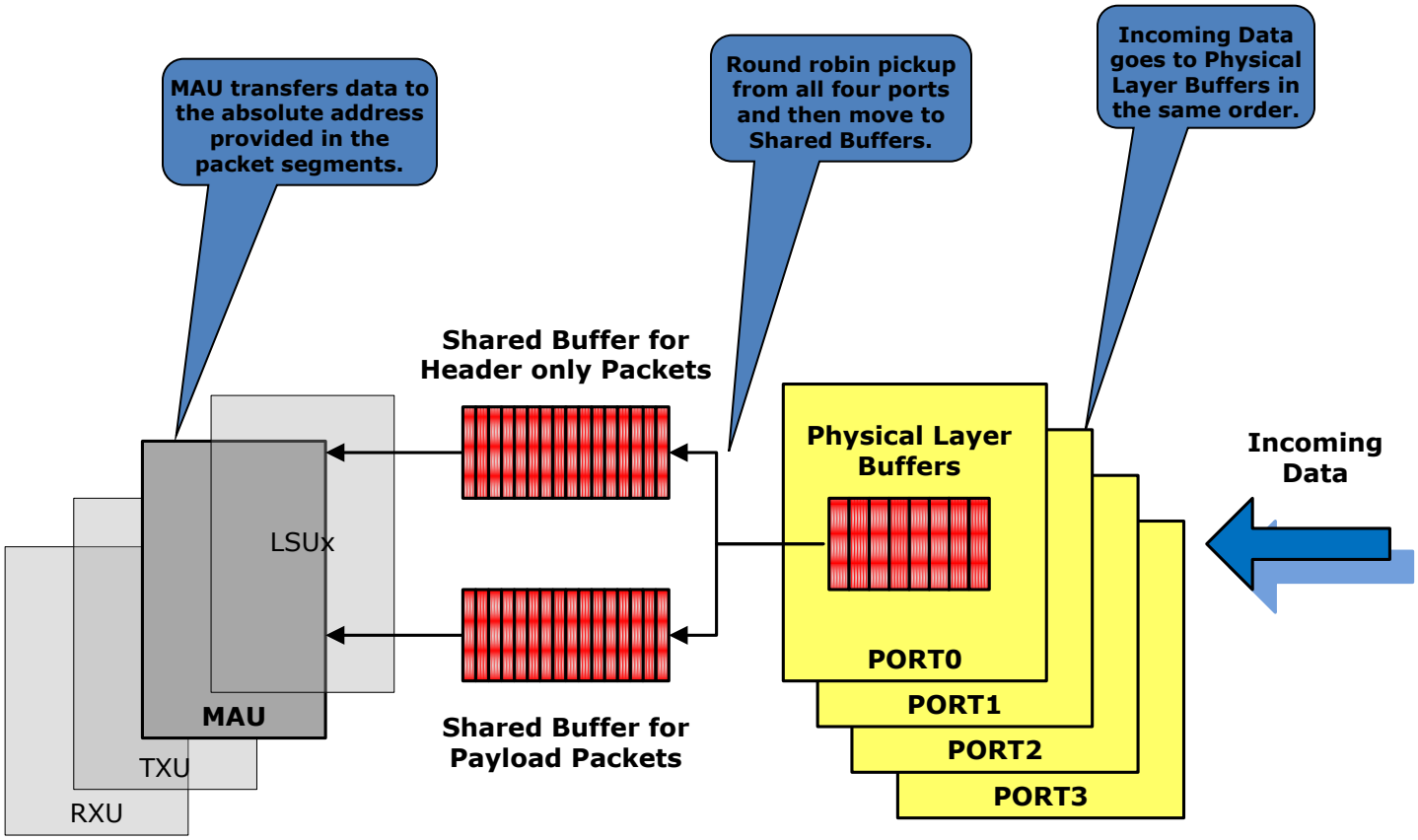
- In this mode, the EDMA programs the shadow registers.
  - The LSUx_EDMA bit is available to program this mode in the SETUP register.
  - The EDMA programs LSU registers Reg0 to Reg5.
  - LSU sends the packet out and the completion generates an interrupt which triggers the EDMA once again.
- The pre-requisite is that the LSU used by EDMA must not be used by any other master:
  - This eliminates the possibility that the LSU becomes busy by another master, so reading the busy bit is not required.
  - EDMA will be able to use only one shadow register so full-bit checking is also not required.
  - So LSU Reg6 read is not required for EDMA mode of operation.

Rx Operation: MAU

- MAU issues four outstanding VBUSM transactions (write/read commands):
  - Those four cannot be (same SrcID) && (same DestID) && (same or lower priority).
  - One of these three requirements must not be matching
- If packet is doorbell, then complete all outstanding transactions and post interrupt.
- If another doorbell comes and previous doorbell is not complete, then a RETRY is attempted on that doorbell.
- Packet Forwarding:
  - Incoming packet is moved to the MAU local buffer first.
  - The packet applies for credit. If it gets the credit, then it will be moved to shared buffer.
  - Forwarding traffic and local traffic mechanisms are separate to avoid conflicts.



Rx Operation: MAU Example



Message Passing Operation

- SRIO Overview
- DirectIO Operation
- Message Passing Operation
- Other RapidIO Features
- Summary

C66x Message Passing Operations Compared to C64x+

- 16 Transmit & 16 Receive Channels

Queue Range	Number of Queues	Purpose
0 to 511	512	Low priority accumulation. These 512 queues are divided into 16 groups, each group with 32 continuous queues. Each group is monitored with one interrupt.
512 to 639	128	AIF Tx queues. Each queue has a dedicated queue pending signal which drives a CDMA Tx channel.
640 to 671	32	PA Tx queues. Each queue has a dedicated queue pending signal which drives a CDMA Tx channel.
672 to 687	16	SRIO Tx queues. Each queue has a dedicated queue pending signal which drives a CDMA Tx channel.
688 to 691	4	FFTC Tx queues. Each queue has a dedicated queue pending signal which drives a CDMA Tx channel.
692 to 703	12	General purpose.
704 to 735	32	High priority accumulation. Each high priority queue can be monitored based on watermark, and each queue has an interrupt signals.
736 to 799	64	Queues with starvation counters readable by the host. Starvation counters increment each time a pop is performed on an empty queue, and reset when the queue is not empty (or when the starvation count is read).
800 to 831	32	QMSS Tx queues. Used for Infrastructure (core to core) DMA copies and notification.
832 to 863	32	Generic queues reserved for traffic shaping, if it is configured in firmware to support this feature.
864 to 8191	7328	General purpose. It is not safe to use queue 8191 however, because some CDMA override functions use 0xFFF in the low 12 bits to specify non-override conditions.

16 dedicated Tx queues for 16 Tx channels

Queues for 16 Rx channels are assigned from this range.

	QMSS	SRIO	PA	FFTC	AIF
Global Control	0x02a6c000	0x02901000	0x02004000	0x021f0200	0x01f17200
Tx Channel Config	0x02a6c400	0x02901400	0x02004400	0x021f0300	0x01f14000
Rx Channel Config	0x02a6c800	0x02901800	0x02004800	0x021f0500	0x01f15000
Tx Scheduler Config	0x02a6cc00	0x02901c00	0x02004c00	0x021f0400	0x01f17000
Rx Flow Config	0x02a6d000	0x02901e00	0x02004e00	0x021f0600	0x01f16000

C66x Message Passing Operations Compared to C64x+

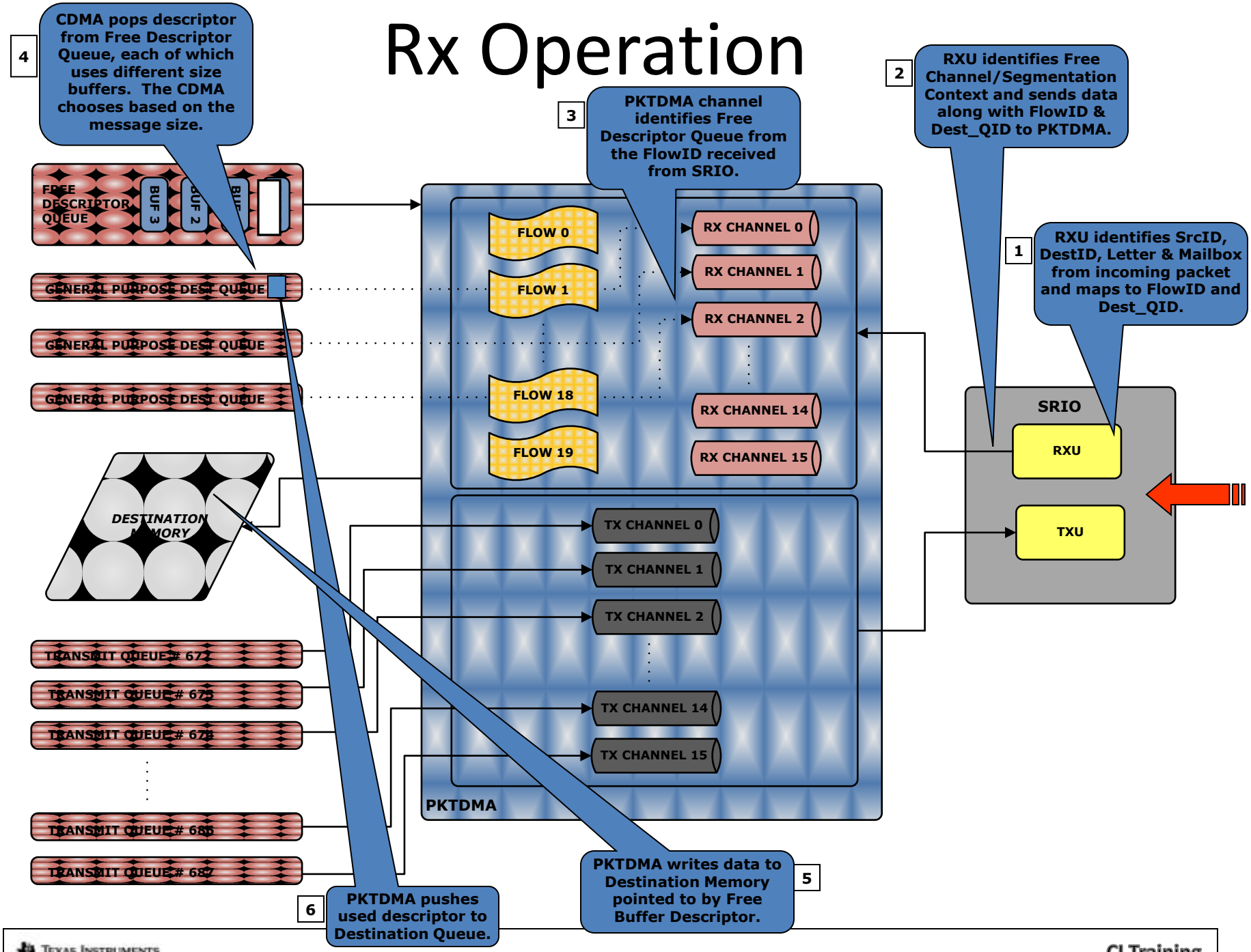
- Maximum 4 KB message size
- Maximum of 16 segments per message
- 64 receive mapping table entries / 32 in 64x+
- 16 outstanding multi-segment + single segment non-posted messages / 4 multi-segment & 12 single-segment in 64x+

Rx Protocol-Specific Part of Descriptor

Word Offset	Bit Fields																													
	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
0	SRC_ID																DEST_ID															
1	Reserved																cc		PRI			tt		LTR		MailBox					

- All fields are same as previous devices. No extra fields have been added here.
- Message size is not part of the SRIO-specific descriptor fields, but instead is located in one of the general descriptor words.
- One Rx descriptor/buffer per Type 11 message

Rx Operation

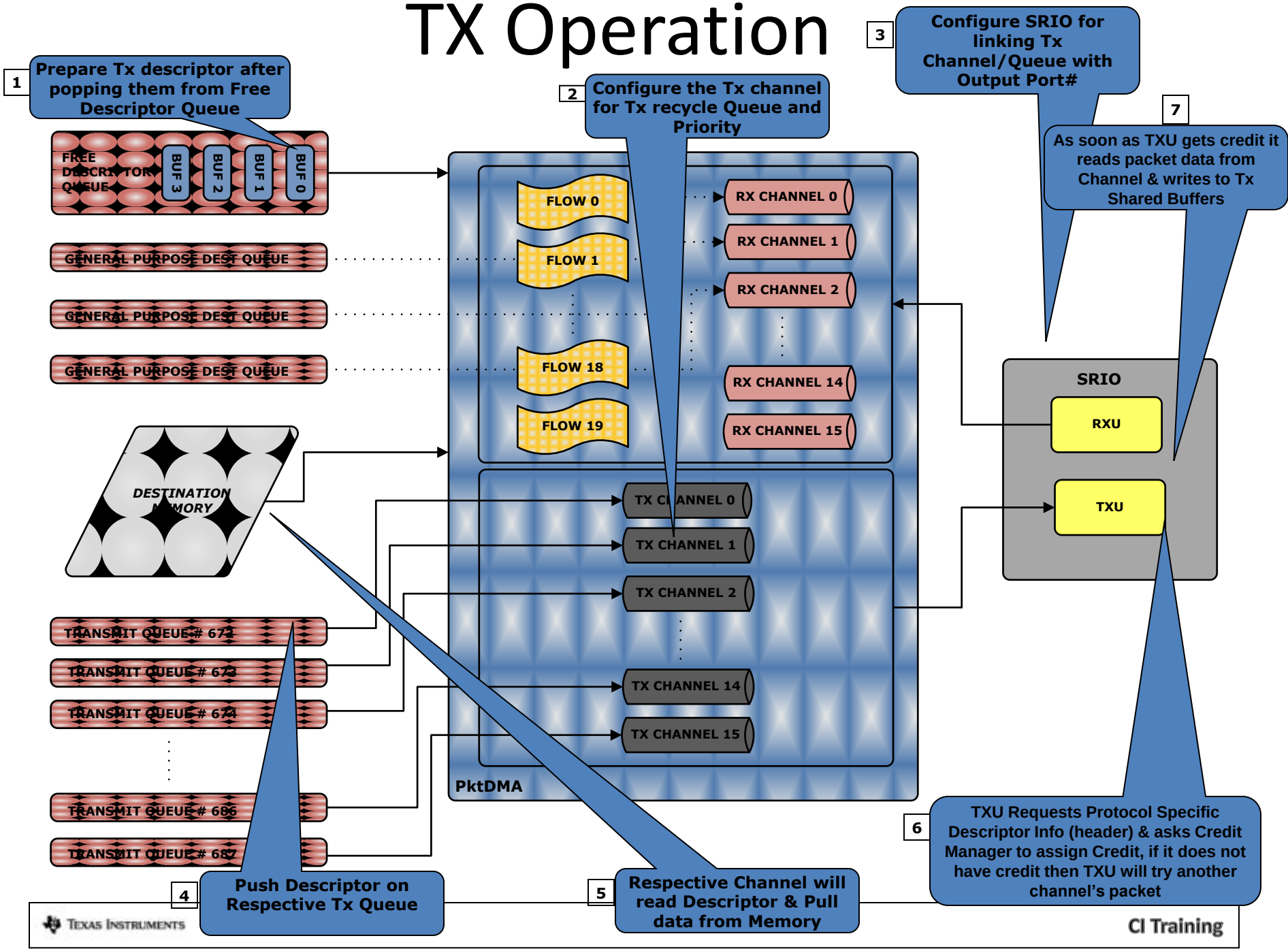


Tx Protocol-Specific Part of Descriptor

Word Offset	Bit Fields																											
	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
0	SRC_ID																DEST_ID															
1	Rsv							Retry_Count				SSIZE					Rsvd						tt		LTR		MailBox					

- All fields are same as previous devices. No extra fields have been added here.
- Message size is not part of the SRIO-specific descriptor fields, but instead is located in one of the general descriptor words.
- One Tx descriptor/buffer per message

TX Operation



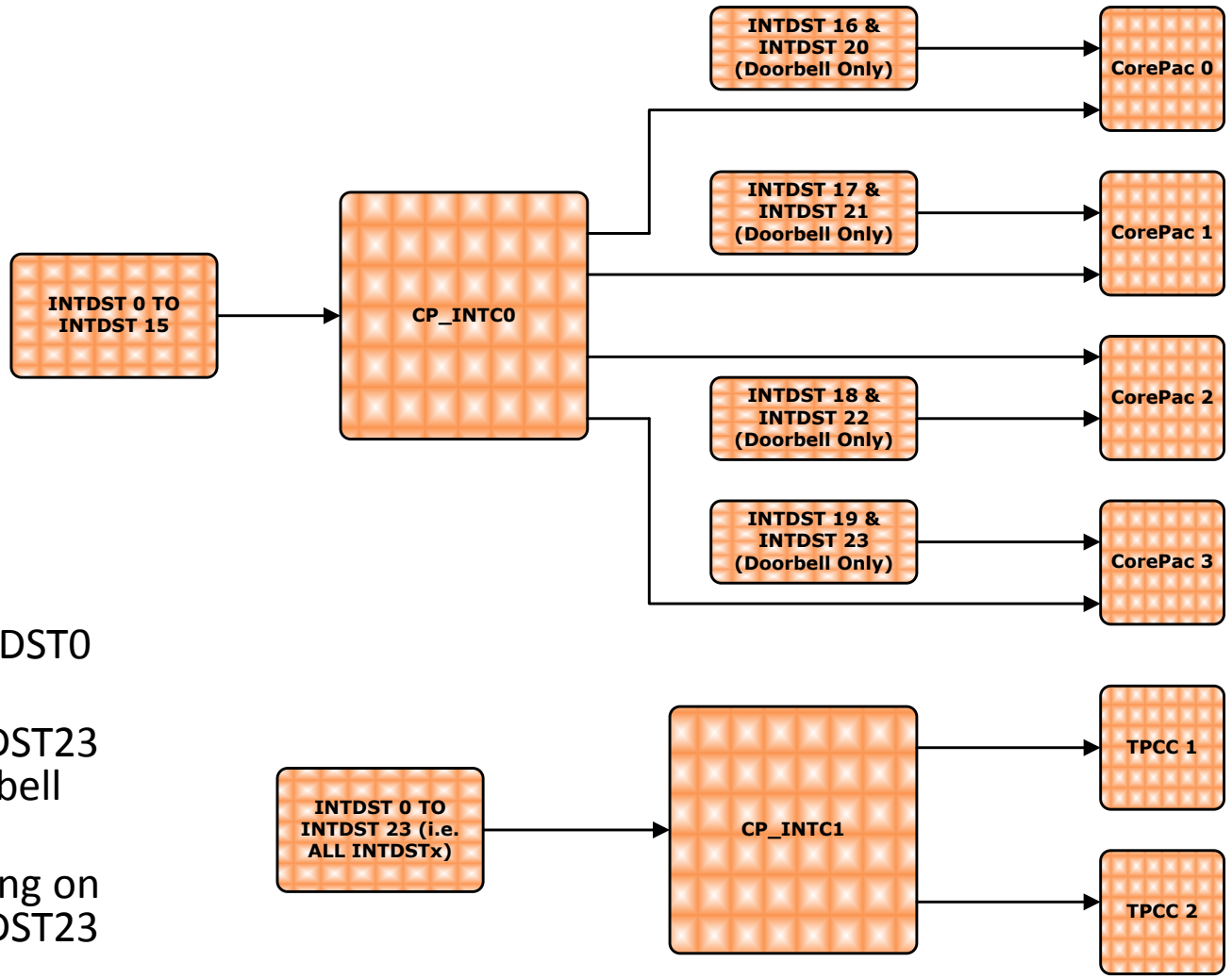
TXU Scheduling

- TXU scheduling requires that a TX queue be dedicated to an outbound port and priority.
- For example:
  - Two active channels/queues
  - One port 4X mode, so all queues are going to the same output port.
  - Queue 1 uses Priority 1 and has something to send.
  - Queue 2 uses Priority 1 + CRF bit and has something to send.
  - Queue 2 is scheduled first and starts to send packets.
  - During the 5th packet segment transfer when the TXU is moving data to the physical layer, TX buffer Queue 3 becomes active and has Priority 2.
  - When Queue 2 is done moving the 5th segment to the physical layer, TXU will begin reading the header info from Queue 3 and subsequently start sending packets from Queue 3.
  - Only after TXU finishes all of the message from Queue 3 will it attempt to go back and send the remaining packets from Queue 2, finally followed by Queue 1 if nothing else of higher priority has shown up in the meantime.

Other RapidIO Features

- SRIO Overview
- DirectIO Operation
- Message Passing Operation
- Other RapidIO Features
- Summary

Interrupt Destinations



- 24 Interrupt Destinations (INTDST0 to INTDST23)
- INTDST16 to INTDST23 are only for Doorbell interrupts
- No Interrupt pacing on INTDST16 to INTDST23

Interrupt Registers

- Three sets of registers route events to INTDSTx:
  - ICSR (Interrupt Condition Set Register)
  - ICCR (Interrupt Condition Clear Register)
  - ICRR (Interrupt Condition Routing Register)
- Doorbell has one special register -- RIO_INTERRUPT_CTL -- which has DBLL_ROUTE bit to decide whether Doorbell ICRR represents (INTDST0 to INTDST15) OR (INTDST16 to INTDST23).
- Doorbell events in Doorbell ICSR are same as previous devices.
- Error events in Error ICSR is same as previous devices.
- LSU events are referenced by SrcID (non-EDMA mode). As KeyStone has 16 local device ID registers, LSU events shows whether a transaction for a particular srcID is complete with Success or Error. Software controlled mapping of srcID error with a particular transaction from a particular LSU is required.
- LSU events with respect to PrivID (EDMA mode)

SRIO LLD

- DirectIO, Type 9 and Type 11 packets support
- APIs With Sequence Of Operations
  - SRIO Peripheral Initialization – Srio_init ()
  - SRIO Driver Instance Initialization – Srio_start ()
    - Initialize Receive & Transmit memory regions for descriptors
    - Creates & Enables CDMA Channels & link them with respective queues
  - SRIO Socket Open – SRIO_sockOpen ()
    - Specify Packet Type
    - Blocking or Non-blocking during Rx Operation
  - Same way SRIO_sockClose ()
  - SRIO Socket Bind – SRIO_sockBind ()
    - Socket Bind is applying RXU mapping entries
    - So max sockets are max RXU mapping entries which is 64
  - Send API – SRIO_sockSend ()
    - Triggers Tx operation with input data pointer, input data size and Destination info (e.g. Mailbox, letter)
  - Receive API – SRIO_sockRecv ()
    - Triggers Rx operation with receive buffer pointer, receive socket

Summary

- C66x SRIO has been enhanced to deliver:
  - Higher performance
  - New transaction types
  - Less required CPU interaction per transaction
  - Better deterministic scheduling
  - More flexibility and system support with increased number of IDs
- For more information:
  - Serial RapidIO (SRIO) for KeyStone Devices User Guide
  - Support forums at the TI E2E Community website