TMS320TCI6487/8 Antenna Interface

User's Guide

Literature Number: SPRUEF4A
December 2006–Revised June 2007

Texas Instruments

Contents

List of Figures

List of Tables

Read This First

About This Manual

This document describes the Antenna Interface module of the TMS320TCI6487/8 device.

Abbreviations

The following abbreviations are used in this manual.

Table 1. Abbreviations Used in the TMS320TCI6487/8 Antenna Interface Guide

Abbreviation	Description	Abbreviation	Description
8b10b	Transmission code for high speed serial links	LUT	Look-up table
AG	Aggregator	LSB	Least significant bit
AIF	Antenna interface	MAC	Media access control
AP	Applications Processor	MAI	Multiple access interference
AxC	Antenna carrier (stream)	MMR	Memory mapped register
AxS	Antenna carrier sample	MSB	Most significant bit
BER	SERDES bit error rate	OBSAI	Open Base Station Architecture Initiative
BFN	UMTS nodeB frame number	PD	Protocol decoder
C&M	Control and management	PE	Protocol encoder
CCM	Control and clock module	PIC	Parallel interface cancellation
CD	Combiner / Decombiner	PLL	Phase lock loop
CFG	Configuration	PS	Packet switched
CI	Common public radio interface input data	RAI	Remote alarm indication
CO	Common public radio interface output data	RAM	Random access memory
CPRI	Common Public Radio Interface	RAC	Receive accelerator co-processor
CRC	Cyclic redundancy check	RF	Radio frequency
CSL	Chip support library	RP1	Reference point 1 (OBSAI)
CW	Control Word (CPRI)	RP3	Reference point 3 (OBSAI)
DB	Data buffer	RSA	Rake search accelerator
DDR2	Dual data rate memory 2	RTL	Register transfer language
DL	Downlink	RX	Receive
DMA	Direct memory access	Rx MAC	Receive media access control
DSP	Digital signal processor	SCR	Switch central resource (VBUS cross bar switch)
EE	Exception event handler	SERDES	Serializer / Deserializer
EMIF	External memory interface	T_c	Chip rate = 1/3.84MHz
FIFO	First in first out queue memory structure	TPDMA	Third party DMA
FDD	Frequency division duplex	TRT	Time reference tick
FSM	Finite state machine	TX	Transmit
GEM	A specific DSP core used throughout TMS320TCI6487/8	Tx MAC	Transmit media access control
HFN	Hyper frame number	UL	Uplink
HW	Hardware	UMTS	Universal mobile telecommunication system
I/F	Interface	VBUSM	Virtual bus multi-issue

Table 1. Abbreviations Used in the TMS320TCI6487/8 Antenna Interface Guide (continued)

Abbreviation	Description	Abbreviation	Description
I/O	Input and output data flow	VBUSP	Virtual bus pipeline
IQ	In-phase and quadrature data	VC	VBUSP configuration bus interface
L2	GEM DSP level 2 SRAM	VD	VBUSP DMA bus interface
LOF	Loss of frame	WCDMA	Wideband code division multiple access
LOS	Loss of signal	XAUI	10-gigabit attachment unit Interface, IEEE 802.3ae
LPSC	Local power sleep controller		

Documentation From Texas Instruments

The following documents describe the TMS320C6000™ devices and related support tools.

SPRU401 — *TMS320C6000 Chip Support Library API Reference Guide.* Describes the TMS320C6000 chip support library (CSL) that is a set of application programming interfaces (APIs) used to configure and control all on-chip peripherals. CSL is intended to make it easier for developers by eliminating much of the tedious work usually needed to get algorithms up and running in a real system.

SPRUEF5— *TMS320TCI6487/8 Frame Synchronization User Guide* Describes the frame synchronization module on the TMS320TCI6487/8 devices.

SPRS358— *TMS320TCI6487/8 Communications Infrastructure Digital Signal Processor Data Manual.* Describes the features of the TMS320TCI6487/8 dcommunications infrastructure digital signal processors (DSPs) and provides pinouts, electrical specifications, and timing information.

1. EDMA 3.0 Specification
2. TMS320TCI6487/8 Antenna Data Servicing Application Note, v0.0.3, WI-FARADAY-APPS-009
3. CSL 3.x Design Specification Document, v1.03, CSL-001-DES
4. TMS320TCI6487/8 SERDES Hardware Design Guidelines

Trademarks

TMS320C6000 is a trademark of Texas Instruments.

All other trademarks are the property of their respective owners.

TMS320TCI6487/8 Antenna Interface

1 Introduction

The TMS320TCI6487/8 antenna interface (AIF) module is a peripheral that supports transfers of baseband IQ data between uplink and downlink baseband DSP processors and a high speed serial interface. The AIF supports both the OBSAI RP3 and CPRI protocols. This chapter describes the top-level architecture, general operation, and features of the antenna interface module.

1.1 Purpose of the Peripheral

The TMS320TCI6487/8 DSP is a high-performance system-on-a-chip (SoC) for the WCDMA-FDD infrastructure baseband solution. Traditional silicon platforms that leading system vendors use in this market consist of C64x DSP + ASICs. To make the SoC a reality, the architecture of the TMS320TCI6487/8 DSP is constructed to include the MAC, symbol rate, transmit chip rate, receive chip rate, the chip rate assist, and the antenna interface functionality. This approach allows you to develop your WCDMA-FDD modem design with an all TMS320TCI6487/8 DSP platform. If you are a system vendor who supports both backplane industry standards of OBSAI and CPRI, you can use the TMS320TCI6487/8 DSP to develop an SoC that allows you to directly drive OBSAI and CPRI backplanes.

1.2 Features

- Supports OBSAI RP3 and CPRI standards
- Supports the following topologies:
  - Star
  - Daisy Chain
  - U Daisy Chain
- 6 SERDES links
- 8b10b line encoding / decoding
- Supports 1x, 2x and 4x serial link rates for each standard
- Independent rates per link
- RP3 message reception based on address and type
- RP3 message transmission based on transmission rules
- RP3 link combining and decombining
- RP3 message insertion
- CPRI bit interleaving and AxC container packing/unpacking
- CPRI Synchronization and L1 inband protocol support
- CPRI support of fast and slow C&M sub-channel
- Link redirection
- Supports over sample ratios:
  - Downlink - 1x
  - Uplink - 2x
- Supports Inter-TMS320TCI6487/8 communications:
  - RP3 packet-switched messages
  - CPRI vendor-specific sub-channel

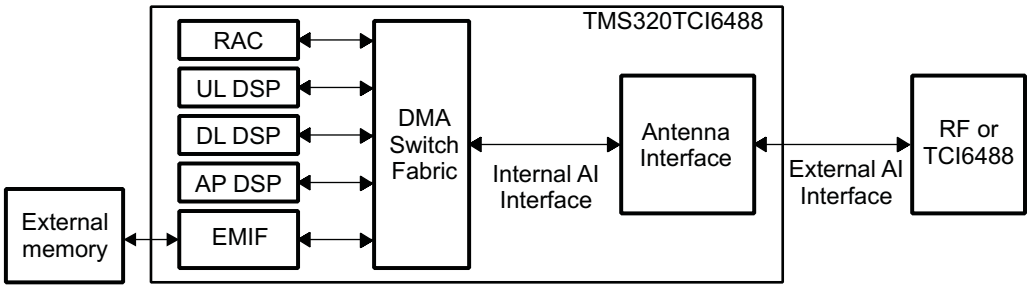
- Antenna data aggregation per AxC
  - 16-bit or 15-bit downlink
  - 8-bit or 7-bit PIC data
  - Automatic saturation (separate I and Q)
- Multiple loopback test modes
- Supports PIC
- Supports data tracing
- Supports clock stop and emulation

2 Overview of Antenna Interface Hardware and Software Components

2.1 Hardware Component

When the antenna interface is in operation, it is static in nature. It continuously receives and transmits antenna data. The needed uplink and/or downlink processing is performed outside of the antenna interface. The uplink subsystem processes the uplink data and the downlink subsystem processes the downlink data. Basically, the antenna interface acts as a buffer so that the system will not lose any samples. The antenna interface supports both OBSAI RP3 and CPRI protocols. A lot of the functionality of the antenna interface is not specific to either protocol. Antenna IQ stream data is the primary transferred data type; and, the inter-TMS320TCI6487/8 control data (or control data to/from the RF units) is the secondary transferred data type.

Figure 1. TMS320TCI6487/8 Antenna Interface Module Overview



The DSP cores that are part of the downlink subsystem are labeled as either 1) DL DSP or 2) TX DSP. The DSP cores that are part of the uplink subsystem are labeled as either 1) UL DSP or 2) RX DSP. You can also make the RAC unit part of the uplink subsystem. The DSP core that is assigned other functions that are not primarily part of either the downlink or uplink subsystems is the application DSP and is labeled the AP DSP. The internal antenna interface connects the AIF with the three DSP cores and the RAC unit via the DMA-switch fabric and control-switch fabric. The AIF internal connection to the switch fabric is the VBUS. The external AIF interface (or SERDES interface) connects the AIF with either RF units and/or other TMS320TCI6487/8 units. A second internal AIF that connects to the configuration-switch fabric is not shown in Figure 1. This document describes an example topology that has one DL DSP, one UL DSP, and one AP DSP.

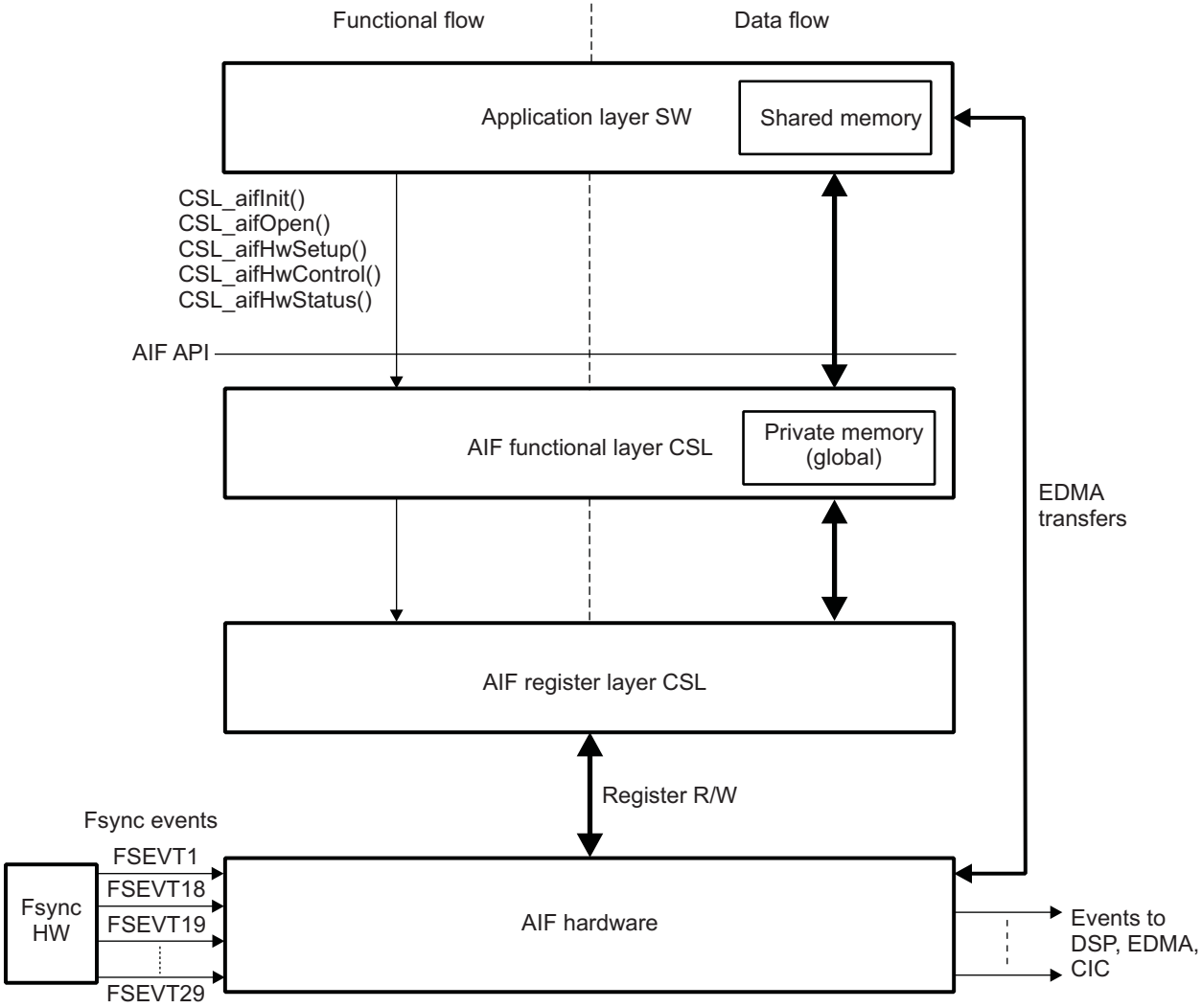
The uplink subsystem receives uplink data from the external interface. The downlink subsystem sends downlink data to the external interface.

2.2 Software Component

Any of the three GEMS cores on the TMS320TCI6487/8 DSP can program the AIF. The organization of software is shown in Figure 2. The advantages of layering the software in this manner are layer-independence and a greater degree of abstraction of the module.

The function flow shown on the left side of the vertical dotted line in Figure 2 mainly constitutes the commands that are used to initialize, configure, retrieve status, and send control commands to the AIF hardware. The data flow shown on the right of the dotted line constitutes the input and output arguments that must be passed between different layers to support configuration, query status of the AIF, etc. The application sets up the EDMA transfers that are shown in Figure 2. The AIF driver layer does not set up any DMA and primarily uses the configuration switch fabric to set up, modify, and query the AIF module.

Figure 2. Overview of Antenna Interface Software Organization



2.2.1 Description of Software Layers

Section 2.2.1.1 through Section 2.2.1.3 provide a brief descriptions of the different software layers.

2.2.1.1 Application Layer

The application layer is the uppermost layer. It constitutes the software that makes API calls to configure, query, and modify the behavior of the AIF module. The application layer is responsible for allocation and de-allocation of memory to store AIF configuration, status, and hardware control information. The application layer has overall control of the AIF module. The application layer software is written in a higher-level language like C and runs on a GEM core.

2.2.1.2 AIF Driver Layer

The AIF driver layer is an abstraction layer that resides between the register-level chip support library and the application code. The AIF driver-layer is written in the C programming language. The AIF layer uses the information that the application gives to make calls to the layer below it (register layer CSL). The AIF driver layer presents a high level of abstraction to the application layer so that you can program the AIF without knowing all of the inner-working details of it. The AIF driver uses private memory (memory used exclusively by the AIF driver layer) to store information that is specific to the antenna interface (such as look-up tables). The private memory that the AIF driver layer requires is allocated at compile time and is declared as a global object. Since the private memory that the AIF requires is global, the compiler creates it in the .bss section.

2.2.1.3 AIF Register Layer

The AIF register layer typically makes read or write operations to one of the memory-mapped registers of the AIF. This software is written in the C programming language.

2.2.2 AIF CSL Limitations

The CSL 3.x specification limits the use of context memory to store state information in the CSL layer (see *CSL 3.x Design Specification Document*). This requirement limits the CSL's ability to check for illegal operations. For example, the configuration of a link without disabling it first is an illegal operation. The CSL must maintain state information in its context to prevent the enabling of a link without configuring it first.

Since the AIF driver layer CSL is required to store transmission look-up tables in order to perform abstraction, private memory is required. However, the AIF CSL does not store any state information.

2.2.3 Functionality of the Software Layer

The Software layer performs the following functions:

- Configuration of individual AIF links
- Enabling and disabling of individual AIF links
- Configuration of frame sync events that are required for the AIF and also the frame sync events which are required to trigger the transfer of data to/from AIF data buffer to/from GEM L1/L2 or DDR memories
- Enabling and disabling of frame sync events
- Configuration of EDMA channels to transfer data to/from AIF data buffer to/from GEM L1/L2 or DDR memories
- Handling of AIF error/exception conditions

3 Introduction to Protocol

The antenna interface supports both the OBSAI RP3 and the CPRI protocols. Most of the functionality of the antenna interface is not specific to either protocol.

3.1 OBSAI RP3 Protocol

The antenna (RP3) interface is the interface for transferring antenna data between baseband chips (TMS320TCI6487/8s) and between baseband chips and RF modules in both the uplink and the downlink directions. The data transferred is of two basic types:

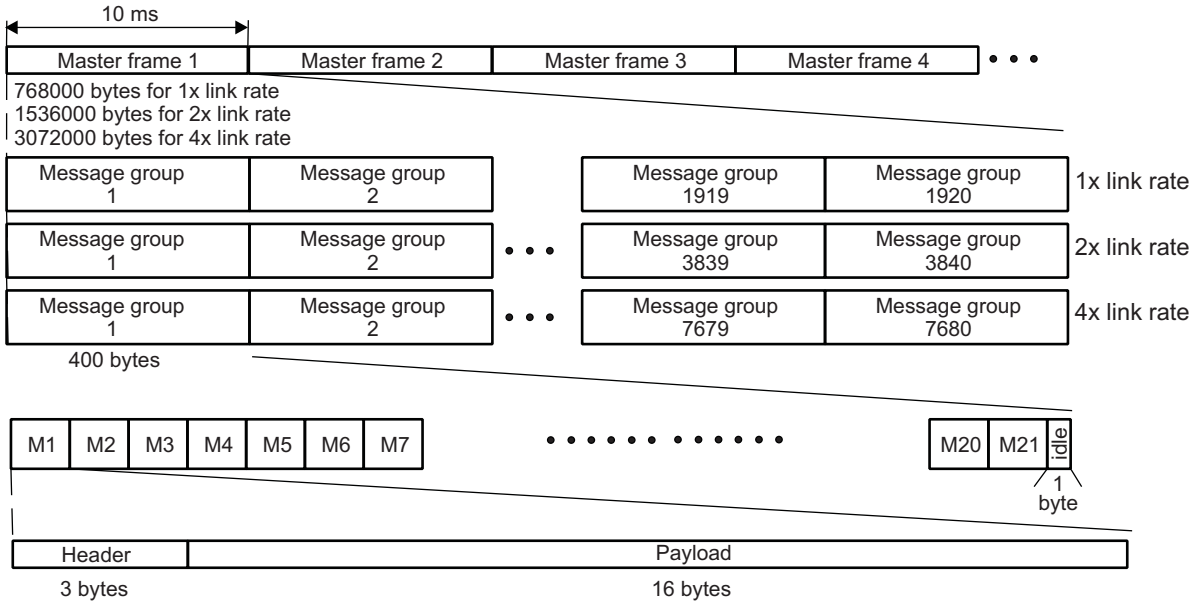
- Circuit switched data consists of antenna IQ data. This data is a constant stream with regular flow characteristics.
- Packet-switched data is control data that is used for communication between TMS320TCI6488 DSP cores and/or RF units. Packet-switched data is event driven with irregular traffic flow characteristics.

The number of bytes in an OBSAI RP3 message group for 1x, 2x, and 4x links is 400. This consists of 20 data message slots (each has 19 bytes), one control message slot (19 bytes) and one idle byte. The physical layer of the bus provides counter values for the data and control message slots. Transmission of messages is done with respect to these counters.

3.1.1 Frame structure

The supported frame structure is illustrated in Figure 3. The master frame consists of 1,920 message groups for 1x link rate, 3840 message groups for 2x link rate and 7680 message groups for 4x link rate. Each message group is further divided into 21 message slots. These are marked with the letter M* in Figure 3. The message group ends at the idle byte, which is a special byte defined in 8b10b-coding used in OBSAI RP3 protocol. These so-called K-characters are unique and can be identified from the serial stream in an unambiguous manner. Therefore, these are used to separate message groups. K28.5 idle byte is used in all other message groups, but the last message group of master frame uses K28.7. The different idle code in the last message group is how the master frames are separated.

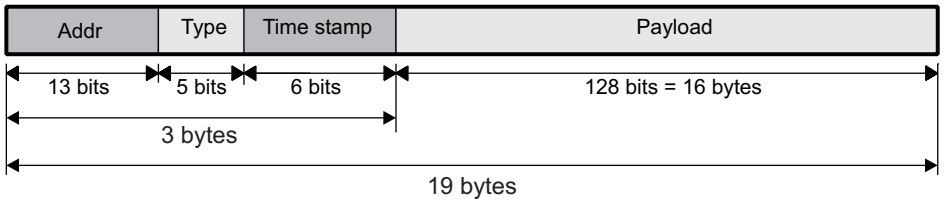
Figure 3. Supported OBSAI RP3 Frame Structure



3.1.2 Packet Descriptions

The antenna interface message packets are specified in the OBSAI RP3 specification. The packet is fixed in length and the structure is depicted in Figure 4.

Figure 4. OBSAI RP3 Protocol Packet Structure



The message fields are described in more detail in the following sections.

3.1.2.1 Address

The address controls the routing of each message. In the downlink direction (from Baseband to RF), all message transfers are point-to-point, and the address identifies the target node. Both multi-casting (point-to-multi-point) and point-to-point message transfers are applied in uplink direction (from RF to Baseband). Uplink antenna sample data as well as some measurement results may require multi-casting; all other message transmissions in uplink direction are typically point-to-point.

The OBSAI RP3 packet contains a 3-byte header. Within those three bytes is the 13-bit address field. TMS320TCI6487/8 uses 10 of those 13 bytes. The 10 bits could be any of the 13 bits.

Reserved addresses

- Address 00000000xxxxx is reserved for initial booting of the bus network.
- Address 1111111111111 is reserved for empty message.

Downlink address

- The address used in downlink streams is based on the antenna carrier destination.
- Since there are 48 downlink streams, 48 different downlink addresses can be assigned.

Uplink address

- The address used in uplink streams is based on the antenna carrier source.
- There can be a maximum of 48 streams received but this does not limit the number of unique addresses to 48. Multiple addresses could target the same stream.

Non-IQ messages

- The address used is the destination.
- Addresses are passed to one of the DSP cores.
- Each address can be directed to any of the three DSP cores.

3.1.2.2 Type

The TYPE field identifies the content of payload data. Table 2 presents the possible payload types. You can use all 5 bits of TYPE to route the message to the appropriate destination. There is no specific support for any type except for WCDMA/FDD. The TYPE field is used to direct the message to the IQ data buffer or the control message FIFO. Messages routed to the IQ data buffer are expected to be in the WCDMA/FDD format regardless of the TYPE field.

Table 2. Content of Type Field

Payload data type	Content of Type field
Control	0000 0000
Measurement	0000 0001
WCDMA/FDD	0000 0010
WCDMA/TDD	0000 0011
GSM/EDGE	0000 0100
TETRA	0000 0101
CDMA2000	0000 0110
WLAN	0000 0111
LOOPBACK	0000 1000
Frame Clock Burst	0000 1001
Ethernet	0000 1010
RTT message	0000 1011
802.16	0000 1100
Virtual HW reset	0000 1101
Currently not in use	0000 1110 - 0001 1111

3.1.2.3 Time Stamp

The time stamp field relates the payload data to a specific time instant.

Consider a block of WCDMA/FDD antenna samples that exist in the payload of a message. In uplink direction, time stamp identifies the time instance when the last sample of the message was available at the output of the channelizer block (down converter, FIR filter). In downlink direction, time stamp defines the time instant when the first sample of the payload must be inserted into the modulator (up converter, FIR filter). Reference time is the WCDMA frame clock of the BTS.

The WCDMA/FDD time stamp is calculated as follows:

TIMESTAMP = (CHIP NUMBER IN SLOT / 4) MOD 64,

Where CHIP NUMBER IN SLOT stands for the chip Index of a WCDMA/FDD time slot. In UL and DL directions, CHIP NUMBER denotes the time Index of the last and first chip (sample) of the message, respectively.

In WCDMA/FDD, 100 frames per second exist and each frame contains 15 time slots. Altogether, 1,500 time slots exist per second while every time slot consists of 2,560 chips indicating that the CHIP NUMBER IN SLOT takes values between 0 and 2,559.

At the end of the master frame, the K28.7 idle byte is sent. This indicates that the next chip is chip number 0. The timestamp for this chip is 0.

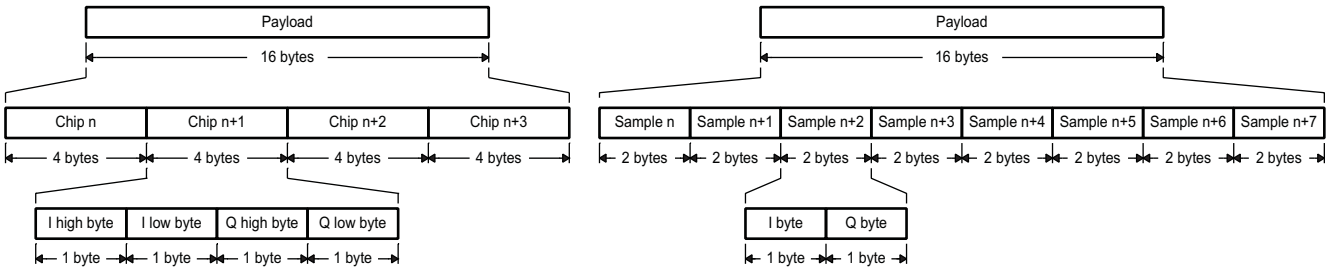
Notes:

- Timestamp always starts at 0 on frame boundaries.
- Timestamp is automatically generated for downlink circuit-switched data.
- Expected timestamp is verified for received data.
- Expected timestamp is compared in the aggregator before aggregation.

3.1.2.4 Payload

You can use the payload of RP3 packet (or message, as it is also called) in two different ways depending on whether the data is UL or DL data. UL data is 8- bit I and Q with an over-sampling ratio of two and DL data is 16-bit I and Q with sampling ratio 1. Therefore, the payload can carry four chips in both cases. This is depicted in Figure 5.

Figure 5. DL and UL Data-Mapping to RP3 Message Payload, Respectively



3.1.3 Transmission Rules

The number of bytes in an OBSAI RP3 message group for 1x, 2x, and 4x links is 400. This consists of 20 data slots (each has 19 bytes), 1 control slot (19 bytes), and 1 IDLE byte. Every (10ms/1,920) 5.21us the number of slots for a 1x, 2x, and 4x links are shown in Table 3. The bus manager provides detailed rules for message transmission for each slot. Rules for data and control messages are provided separately. The physical layer of the bus provides counter values for the data and control message slots. Transmission of messages is done with respect to these counters.

Transmission rules are only used for message transmission and not message reception.

Table 3. Number of Slots and IDLE Bytes in 5.21us

Link rate	Number of Data Slots	Number of Control Slots	Number of IDLE Bytes
1x	20	1	1
2x	40	2	2
4x	80	4	4

The master frame length and UMTS frame length is 10ms. There are 1,920 message groups at 768Mbps, 3,840 message groups at 1,536Mbps, and 7,680 message groups at 3,072Mbps link rate.

Message slots for each path are specified by the pair of numbers (Index (I), Modulo (M)) so that the equation MessageSlotCounter X=I holds. The number of data messages in a frame is equal to i×N_MG×(M_MG×19+K_MG) and M_MG is the number of message slots in a message group and K_MG is the number of IDLE bytes at the end of a message group. N_MG is the number of message groups in a master frame. N_MG=1920, M_MG=21, and K_MG=1.

Transmission rules:

- Number of data and control slots is shown in Table 3.
- Data slots can be circuit-switched or packet-switched data.
- Supported modulos
  - Circuit-switched slots supports modulos of 4, 8, or 16.
    - A 1x link supports a modulo of 4.
    - A 2x link supports a modulo of 8.
    - A 4x link supports a modulo of 16.
  - Packet-switched messages in data slots can have modulo values of 1, 2, 4, 5, 8, 10, 16, 20, 40, 80, and any 2N*80 multiple. Packet-switched data in control message slots can use any modulo value.
  - Packet-switched slots supports a once-per-frame modulo.
    - A 1x link supports a 1,920 modulo.
    - A 2x link supports a 3,840 modulo.
    - A 4x link supports a 7,680 modulo.
- Control slots can only occur at the end of a message group

An example is used to highlight some of the transmission rules with a single example. This example is shown in Figure 6 with the rules defined in Table 4.

- The antenna carriers are labeled with an A and occur in a data message slot.
- The control messages are labeled with a C and occur in a control message slot.
- Control messages that occur in a data message slot are labeled with a P.
- Empty messages are labeled with an E.

Although message reception is address-based, combining, decombining, and aggregation are positional based, and as such can only support a limited number of transmission rules. Please refer to the respective sections of this document for details.

Figure 6. Transmission Rules Example for a 4x Link

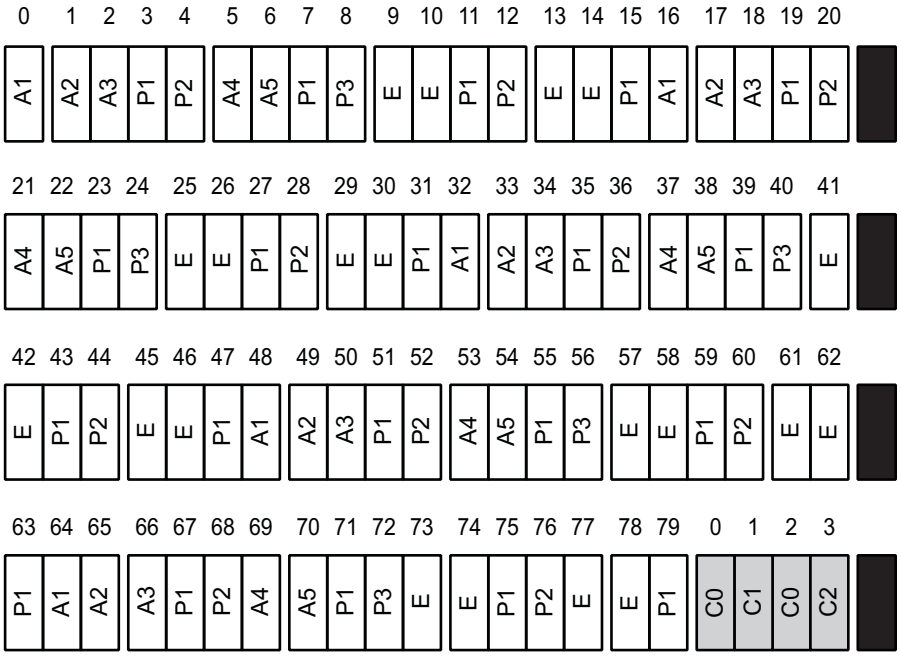


Table 4. Transmission Rule Example

Reference	Modulo (M)	Index (I)	Type	Carrier
Data slot	16	0	Circuit	A1 – 1x BW
Data slot	16	1	Circuit	A2 – 1x BW
Data slot	16	2	Circuit	A3 – 1x BW
Data slot	16	5	Circuit	A4 – 1x BW
Data slot	16	6	Circuit	A5 – 1x BW
Data slot	4	3	Packet	P1 – 4x BW
Data slot	8	4	Packet	P2 – 2x BW
Data slot	16	8	Packet	P3 – 1x BW
Data slot	2	0	Packet	C0
Data slot	4	1	Packet	C1
Data slot	4	3	Packet	C2

3.1.4 Transmission Rule Usage

The general definition of transmission rules gives greater than practical flexibility.

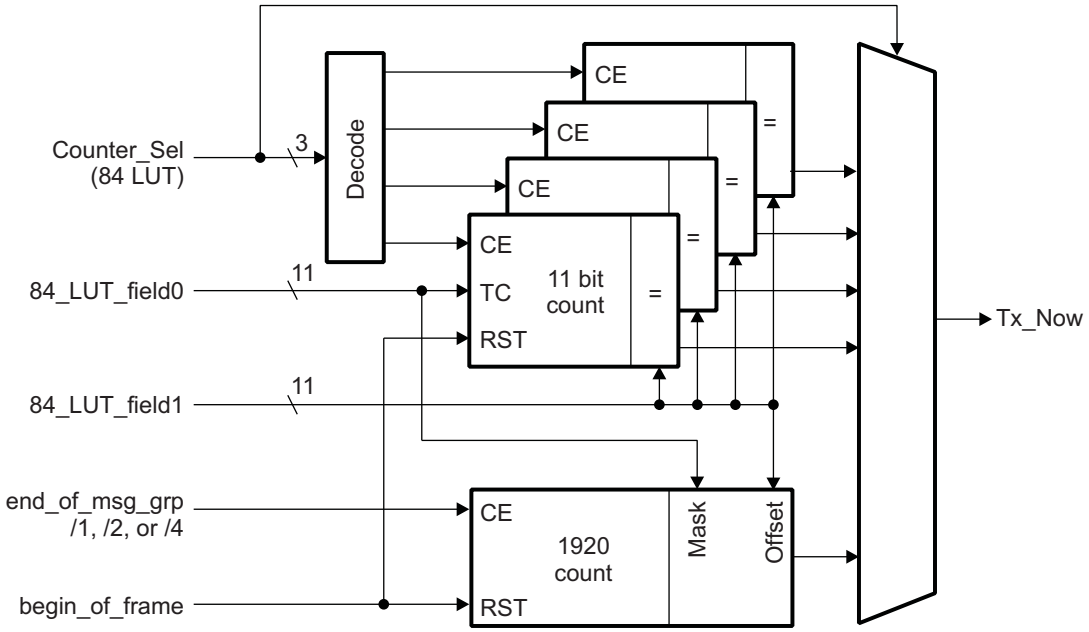
The antenna interface imposes some practical limitations for antenna interface usage. They are as follows:

- The periodicity of transmission rules may not exceed one UMTS frame. This is an OBSAI RP3 specified limitation as well.
- 4x and 2x link rates have multiple sets of control slots. Each set of control slots is treated like an independent lx link set of control slots.
- Transmission rules for message slots must use Modulo (M) values 1, 2, 4, 5, 8, 10, 16, 20, 40, 80, or any 2N*80 multiple.
- Circuit-switched data streams are UMTS rate and transmission rules must be used in a way that this is not violated (this is a testability limitation only).

Note: Antenna interface transmission rule formats do differ significantly from OBSAI RP3 transmission rule formats; however, other than the limitations listed above, they have identical functionality.

Two sets of look-up tables control each link. One look-up table has 21 entries (see Table 114) and the other table has 84 entries (see Table 112). For information about these two look-up tables, see Section 7.1.4.1.

Figure 7. Transmission Rule Generation



The protocol encoder uses transmission rules in a different form than specified in the *TMS320TCI6487/8 Chip Support Library API Reference Guide*; the protocol encoder's format for control slots and data slots differs as well.

3.2 CPRI Protocol

The AIF supports CPRI protocol. The type of protocol, either OBSAI RP3 or CPRI, is selected at boot time and cannot be changed without a re-boot.

3.2.1 Supported CPRI Options

- Topologies:
  - Star
  - Daisy Chain
  - U Daisy Chain (This can be used for redundancy only)
- Aggregation:
  - Sums IQ data of received downlink AxC with locally generated AxC
  - Saturates up to 15 or 16 bits for downlink data
  - Saturates up to 7 or 8 bits for PIC data
- Antenna carriers supported per link
  - Downlink :
    - 3 AxCs (1x link rate, 16-bit sample width, packed (W/R bits at end of basic frame))
    - 4 AxCs (1x link rate, 15-bit sample width, packed)
    - 7 AxCs (2x link rate, 16-bit sample width, packed (W/R bits at end of basic frame))
    - 8 AxCs (2x link rate, 15-bit sample width, packed)
    - 15 AxCs (4x link rate, 16-bit sample width, packed (W/R bits at end of basic frame))
    - 16 AxCs (4x link rate, 15-bit sample width, packed)
  - Uplink:
    - 3 AxCs (1x link rate, 8 bit sample width, packed (W/R bits at end of basic frame))
    - 4 AxCs (1x link rate, 7 bit sample width, flexible position (W/R bits after every AxC container))

- 7 AxCs (2x link rate, 8bit sample width, packed (W/R bits after every AxC container))
- 8 AxCs (2x link rate, 7 bit sample width, flexible position (W/R bits after every AxC container))
- 15 AxCs (4x link rate, 8 bit sample width, packed (W/R bits after every AxC container))
- 16 AxCs (4x link rate, 7 bit sample width, flexible position (W/R bits after every AxC container))

3.2.2 Basic Frame Structure

The basic frame structure consists of 1 control word followed by 15 data words as shown in Figure 8. For a 1x link, the word size is one byte (Y = 0) as shown in Figure 8. For a 2x link, the word size is two bytes (Y = 0 and 1), as shown in Figure 9 and for a 4x rate link, the word size is four bytes (Y=0, 1, 2, and 3), as shown in Figure 10. Y is the byte index within each word. Each basic frame consists of N AxC containers, where N is the number of antenna carriers of 1 chip's worth of IQ data that will fit in a given link.

An AxC container consists of the following:

- From LSB (I_0, Q_0) to MSB (I_{-1}, Q_{-1}) or ($I_{M'-1}$, $Q_{M'-1}$)
- I and Q sample bits are interleaved
- In chronological order
- Consecutively without any reserved bits (R) in between
- Each AxC container is sent as a block
- Overlap of AxC containers is not allowed. The position of each AxC container in the IQ data block is decided by one of the following options
  - Packed position: Each AxC container is sent consecutively without any reserved bits in between and in ascending order of AxC number. Reserved bits can be added at the end of the basic frame to fill the unused bits.
  - Flexible position: For each AxC container, the application decides at what address (W,B - for W>0) in the IQ block the first bit of the AxC container is positioned.
- The supported packing options are shown in the figures in this section. The following is the defined nomenclature for CPRI.
  - B is the bit index (0 to 7).
  - W is word index (0 to 15).
  - Y is the byte index within a word (0 to 3).
  - X is the basic frame number (0 to 255).
  - Z is the hyper-frame number (0 to 149).
  - Z.X.Y is used to define control words.
- The supported lengths for CPRI:
  - The length of a basic frame is 260.4167ns.
  - There are 256 basic frames in a HFN = 66.67us.
  - There are 150 HFN in a BFN = 10ms.

Figure 8. CPRI Basic Frame Structure for 1x Link

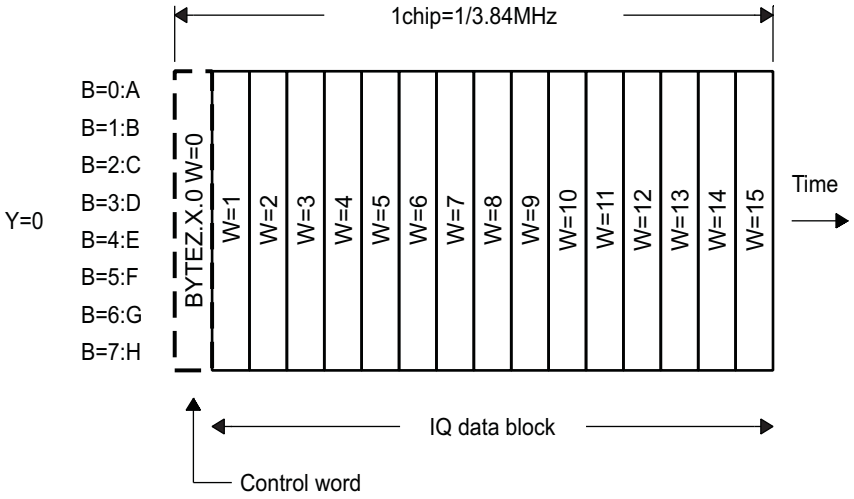


Figure 9. CPRI Basic Frame Structure for 2x Link

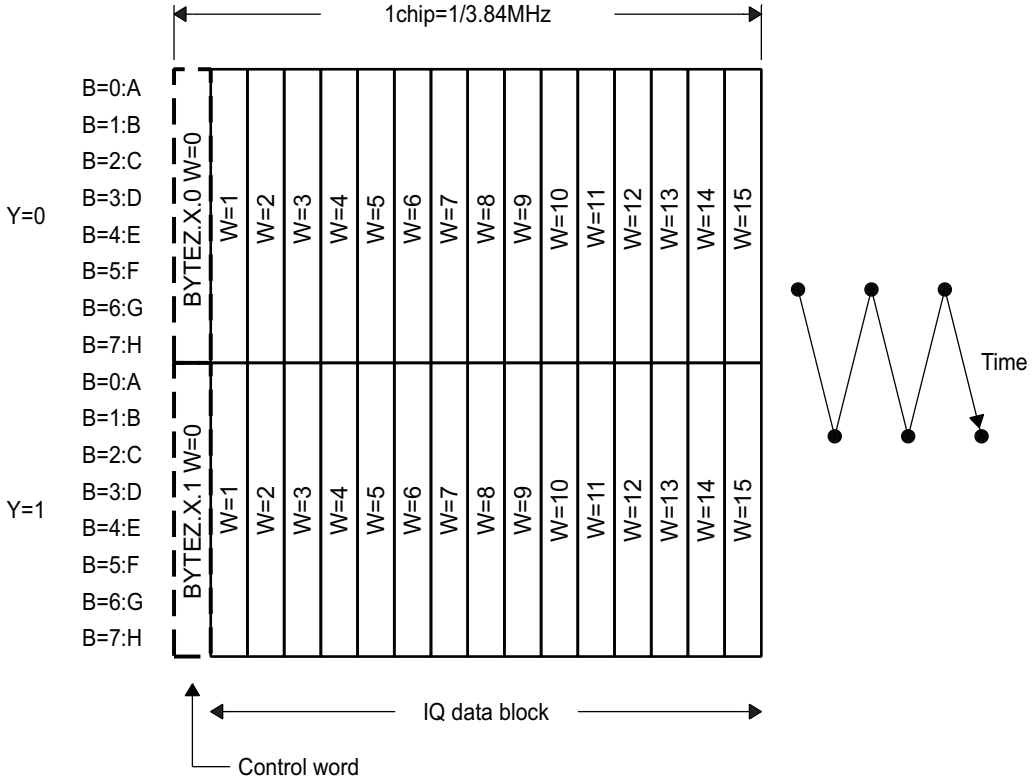
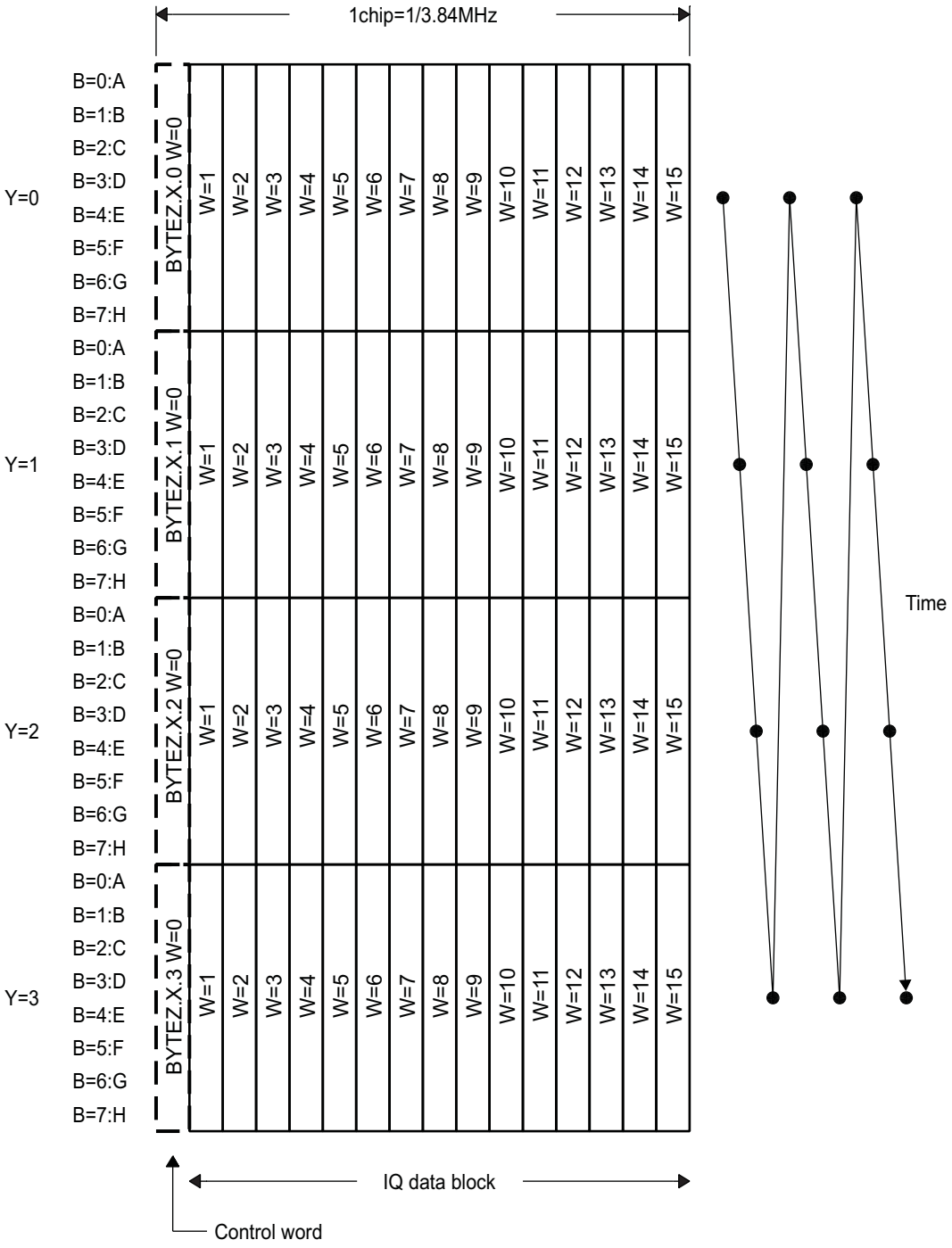


Figure 10. CPRI Basic Frame Structure for 4x Link



3.2.2.1 IQ Sample Width

The supported sample widths are:

- DL - Direction: M = 15 or 16 bits
- UL - Direction: M = 7 or 8 bits
- Only 7/15 or 8/16 bit width combinations are supported.

3.2.2.2 Mapping of IQ Sample Within One AxC Container

An AxC container contains the IQ samples of one AxC for one chip. The supported mapping of IQ samples within an AxC container is as follows:

- DL - Over sampling Ratio: 1
- DL Symbols for IQ samples: I,Q
- UL - Over sampling Ratio: 2
- UL Symbols for IQ samples: I,Q,I',Q'

3.2.2.3 Mapping of AxC Container within One Basic Frame

The CPRI stream format interleaves IQ data on a bit-per-bit basis.

Table 5 lists the references for all of the link rates vs. 15/7 and 16/8 bit sizes.

Table 5. Link Rate vs. DL/UL Bit Size Reference Table

Link Rate	DL and/or UL Bit Size	Table
1x	15/7	CPRI Data Format (1x link - 15 bit DL, 7 bit UL)
2x	15/7	CPRI Data Format (2x link - 15 bit DL, 7 bit UL)
4x	DL=15	CPRI Data Format (4x link - 15 bit DL)
4x	UL=7	CPRI Data Format (4x link - 7 bit UL)
1x	16/8	CPRI Data Format (1x link - 16 bit DL, 8 bit UL)
1x	16/8	CPRI Data Format (2x link - 16 bit DL, 8 bit UL)
4x	DL=16	CPRI Data Format (4x link - 16 bit DL)
4x	UL=8	CPRI Data Format (4x link - 8 bit UL)

The supported mapping of AxC container within one basic frame is listed in Table 6.

Table 6. Supported IQ Sample Widths

Direction	Bits	Packing Option	Packing Description
Downlink	16	Packed	Read/write bits at the end of basic frame
Downlink	15	Packed	No read bits
Uplink	8	Packed	Read/write bits at the end of basic frame
Uplink	7	Flexible	Read/write: read after every AxC container

The (W,B) mapping of the starting location for each AxC container for 7-bit UL data is shown in Table 7.

Table 7. (W,B) Address of the First Bit of the AxC Containers for 7-Bit UL

AxC #	1x Link Rate	2x Link Rate	4x Link Rate
0	(0, 0)	(0, 0)	(0, 0)
1	(3, 6)	(1, 14)	(0, 30)
2	(7, 4)	(3, 12)	(1, 28)
3	(11, 2)	(5, 10)	(2, 26)
4	-	(7, 8)	(3, 24)
5	-	(9, 6)	(4, 22)
6	-	(11, 4)	(5, 20)
7	-	(13, 3)	(6, 18)
8	-	-	(7, 16)

Table 7. (W,B) Address of the First Bit of the AxC Containers for 7-Bit UL (continued)

AxC #	1x Link Rate	2x Link Rate	4x Link Rate
9	-	-	(8, 14)
10	-	-	(9, 12)
11	-	-	(10, 10)
12	-	-	(11, 8)
13	-	-	(12, 6)
14	-	-	(13, 4)
15	-	-	(14, 2)

Figure 11. CPRI Data Format (1x link - 15 bit DL, 7 bit UL)

	CW	A0			A1				A2				A3			
		16 words = 16 bytes														
LSB	Control word	I0,0	I0,4	I0,8	I0,12	I1,1	I1,5	I1,9	I1,13	I2,2	I2,6	I2,10	I2,14	I3,3	I3,7	I3,11
		Q0,0	Q0,4	Q0,8	Q0,12	Q1,1	Q1,5	Q1,9	Q1,13	Q2,2	Q2,6	Q2,10	Q2,14	Q3,3	Q3,7	Q3,11
		I0,1	I0,5	I0,9	I0,13	I1,2	I1,6	I1,10	I1,14	I2,3	I2,7	I2,11	I3,0	I3,4	I3,8	I3,12
DL		Q0,1	Q0,5	Q0,9	Q0,13	Q1,2	Q1,6	Q1,10	Q1,14	Q2,3	Q2,7	Q2,11	Q3,0	Q3,4	Q3,8	Q3,12
		I0,2	I0,6	I0,10	I0,14	I1,3	I1,7	I1,11	I2,0	I2,4	I2,8	I2,12	I3,1	I3,5	I3,9	I3,13
		Q0,2	Q0,6	Q0,10	Q0,14	Q1,3	Q1,7	Q1,11	Q2,0	Q2,4	Q2,8	Q2,12	Q3,1	Q3,5	Q3,9	Q3,13
		I0,3	I0,7	I0,11	I1,0	I1,4	I1,8	I1,12	I2,1	I2,5	I2,9	I2,13	I3,2	I3,6	I3,10	I3,14
MSB		Q0,3	Q0,7	Q0,11	Q1,0	Q1,4	Q1,8	Q1,12	Q2,1	Q2,5	Q2,9	Q2,13	Q3,2	Q3,6	Q3,10	Q3,14

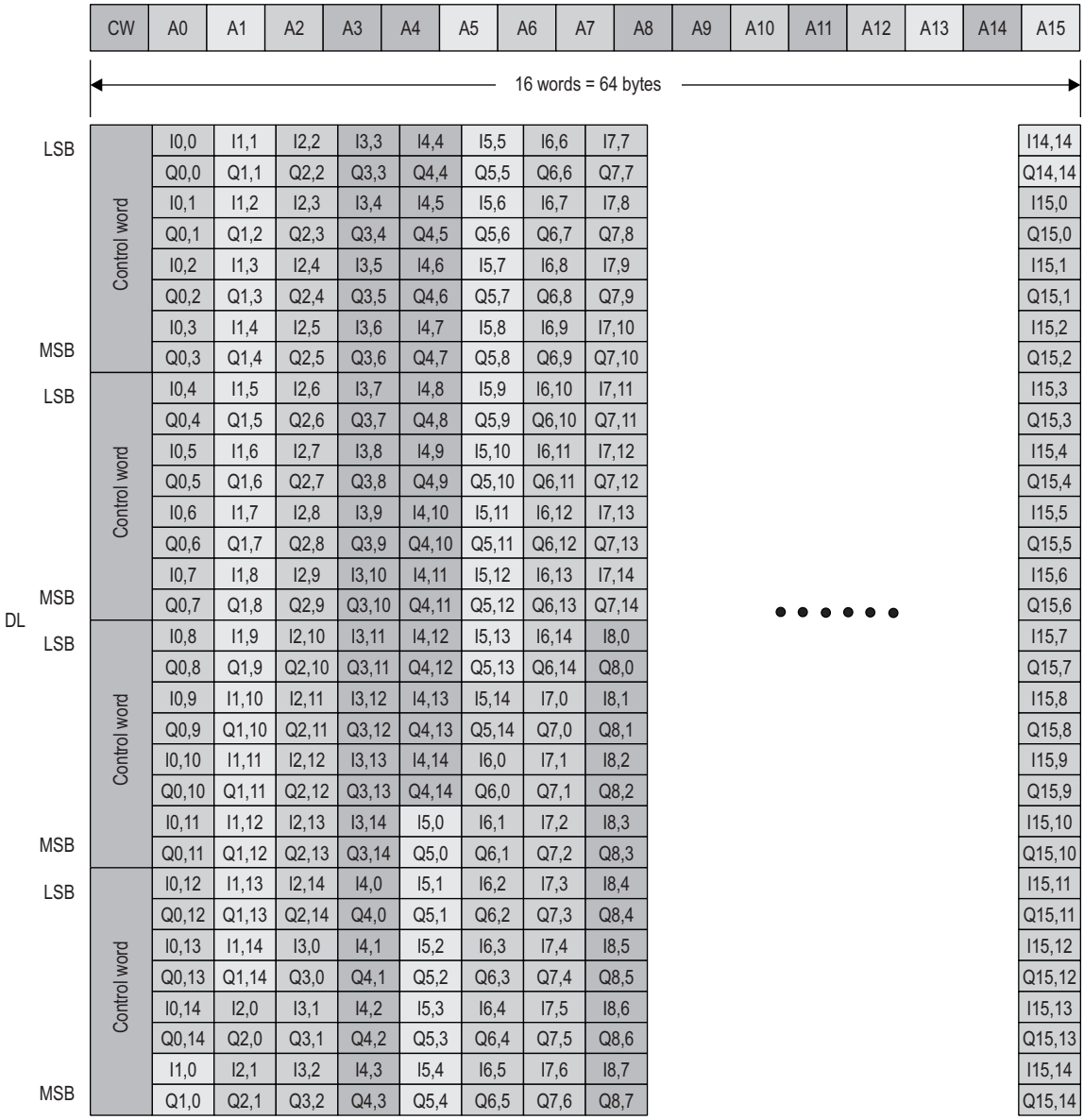
LSB	Control word	I0,0	I0,4	I'0,1	I'0,5	I1,1	I1,5	I'1,2	I'1,6	I2,2	I2,6	I'2,3	r	I3,3	I'3,0	I'3,4
		Q0,0	Q0,4	Q'0,1	Q'0,5	Q1,1	Q1,5	Q'1,2	Q'1,6	Q2,2	Q2,6	Q'2,3	r	Q3,3	Q'3,0	QI'3,4
		I0,1	I0,5	I'0,2	I'0,6	I1,2	I1,6	I'1,3	r	I2,3	I'2,0	I'2,4	I3,0	I3,4	I'3,1	I'3,5
UL		Q0,1	Q0,5	Q'0,2	Q'0,6	Q1,2	Q1,6	Q'1,3	r	Q2,3	Q'2,0	Q'2,4	Q3,0	Q3,4	Q'3,1	Q'3,5
		I0,2	I0,6	I'0,3	r	I1,3	I'1,0	I'1,4	I2,0	I2,4	I'2,1	I'2,5	I3,1	I3,5	I'3,2	I'3,6
		Q0,2	Q0,6	Q'0,3	r	Q1,3	Q'1,0	Q'1,4	Q2,0	Q2,4	Q'2,1	Q'2,5	Q3,1	Q3,5	Q'3,2	Q'3,6
		I0,3	I'0,0	I'0,4	I1,0	I1,4	I'1,1	I'1,5	I2,1	I2,5	I'2,2	I'2,6	I3,2	I3,6	I'3,3	r
MSB		Q0,3	Q'0,0	Q'0,4	Q1,0	Q1,4	Q'1,1	Q'1,5	Q2,1	Q2,5	Q'2,2	Q'2,6	Q3,2	Q3,6	Q'3,3	r

Figure 12. CPRI Data Format (2x link - 15 bit DL, 7 bit UL)

		CW	A0	A1	A2	A3	A4	A5	A6	A7
			16 words = 32 bytes							
DL	LSB	Control word	I0,0	I0,8	I1,1	I1,9	I2,2	I2,10	I3,3	I3,11	I4,4	I4,12	I5,5	I5,13	I6,6	I6,14	I7,7
			Q0,0	Q0,8	Q1,1	Q1,9	Q2,2	Q2,10	Q3,3	Q3,11	Q4,4	Q4,12	Q5,5	Q5,13	Q6,6	Q6,14	Q7,7
			I0,1	I0,9	I1,2	I1,10	I2,3	I2,11	I3,4	I3,12	I4,5	I4,13	I5,6	I5,14	I6,7	I7,0	I7,8
			Q0,1	Q0,9	Q1,2	Q1,10	Q2,3	Q2,11	Q3,4	Q3,12	Q4,5	Q4,13	Q5,6	Q5,14	Q6,7	Q7,0	Q7,8
			I0,2	I0,10	I1,3	I1,11	I2,4	I2,12	I3,5	I3,13	I4,6	I4,14	I5,7	I6,0	I6,8	I7,1	I7,9
			Q0,2	Q0,10	Q1,3	Q1,11	Q2,4	Q2,12	Q3,5	Q3,13	Q4,6	Q4,14	Q5,7	Q6,0	Q6,8	Q7,1	Q7,9
			I0,3	I0,11	I1,4	I1,12	I2,5	I2,13	I3,6	I3,14	I4,7	I5,0	I5,8	I6,1	I6,9	I7,2	I7,10
	MSB		Q0,3	Q0,11	Q1,4	Q1,12	Q2,5	Q2,13	Q3,6	Q3,14	Q4,7	Q5,0	Q5,8	Q6,1	Q6,9	Q7,2	Q7,10
	LSB	Control word	I0,4	I0,12	I1,5	I1,13	I2,6	I2,14	I3,7	I4,0	I4,8	I5,1	I5,9	I6,2	I6,10	I7,3	I7,11
			Q0,4	Q0,12	Q1,5	Q1,13	Q2,6	Q2,14	Q3,7	Q4,0	Q4,8	Q5,1	Q5,9	Q6,2	Q6,10	Q7,3	Q7,11
			I0,5	I0,13	I1,6	I1,14	I2,7	I3,0	I3,8	I4,1	I4,9	I5,2	I5,10	I6,3	I6,11	I7,4	I7,12
			Q0,5	Q0,13	Q1,6	Q1,14	Q2,7	Q3,0	Q3,8	Q4,1	Q4,9	Q5,2	Q5,10	Q6,3	Q6,11	Q7,4	Q7,12
			I0,6	I0,14	I1,7	I2,0	I2,8	I3,1	I3,9	I4,2	I4,10	I5,3	I5,11	I6,4	I6,12	I7,5	I7,13
			Q0,6	Q0,14	Q1,7	Q2,0	Q2,8	Q3,1	Q3,9	Q4,2	Q4,10	Q5,3	Q5,11	Q6,4	Q6,12	Q7,5	Q7,13
			I0,7	I1,0	I1,8	I2,1	I2,9	I3,2	I3,10	I4,3	I4,11	I5,4	I5,12	I6,5	I6,13	I7,6	I7,14
	MSB		Q0,7	Q1,0	Q1,8	Q2,1	Q2,9	Q3,2	Q3,10	Q4,3	Q4,11	Q5,4	Q5,12	Q6,5	Q6,13	Q7,6	Q7,14

UL	LSB	Control word	I0,0	I0,1	I1,1	I1,2	I2,2	I2,3	I3,3	I3,4	I4,4	I4,5	I5,5	I5,6	I6,6	r	I7,0
			Q0,0	Q0,1	Q1,1	Q1,2	Q2,2	Q2,3	Q3,3	Q3,4	Q4,4	Q4,5	Q5,5	Q5,6	Q6,6	r	Q7,0
			I0,1	I0,2	I1,2	I1,3	I2,3	I2,4	I3,4	I3,5	I4,5	I4,6	I5,6	r	I6,0	I7,0	I7,1
			Q0,1	Q0,2	Q1,2	Q1,3	Q2,3	Q2,4	Q3,4	Q3,5	Q4,5	Q4,6	Q5,6	r	Q6,0	Q7,0	Q7,1
			I0,2	I0,3	I1,3	I1,4	I2,4	I2,5	I3,5	I3,6	I4,6	r	I5,0	I6,0	I6,1	I7,1	I7,2
			Q0,2	Q0,3	Q1,3	Q1,4	Q2,4	Q2,5	Q3,5	Q3,6	Q4,6	r	Q5,0	Q6,0	Q6,1	Q7,1	Q7,2
			I0,3	I0,4	I1,4	I1,5	I2,5	I2,6	I3,6	r	I4,0	I5,0	I5,1	I6,1	I6,2	I7,2	I7,3
	MSB		Q0,3	Q0,4	Q1,4	Q1,5	Q2,5	Q2,6	Q3,6	r	Q4,0	Q5,0	Q5,1	Q6,1	Q6,2	Q7,2	Q7,3
	LSB	Control word	I0,4	I0,5	I1,5	I1,6	I2,6	r	I3,0	I4,0	I4,1	I5,1	I5,2	I6,2	I6,3	I7,3	I7,4
			Q0,4	Q0,5	Q1,5	Q1,6	Q2,6	r	Q3,0	Q4,0	Q4,1	Q5,1	Q5,2	Q6,2	Q6,3	Q7,3	Q7,4
			I0,5	I0,6	I1,6	r	I2,0	I3,0	I3,1	I4,1	I4,2	I5,2	I5,3	I6,3	I6,4	I7,4	I7,5
			Q0,5	Q0,6	Q1,6	r	Q2,0	Q3,0	Q3,1	Q4,1	Q4,2	Q5,2	Q5,3	Q6,3	Q6,4	Q7,4	Q7,5
			I0,6	r	I1,0	I2,0	I2,1	I3,1	I3,2	I4,2	I4,3	I5,3	I5,4	I6,4	I6,5	I7,5	I7,6
			Q0,6	r	Q1,0	Q2,0	Q2,1	Q3,1	Q3,2	Q4,2	Q4,3	Q5,3	Q5,4	Q6,4	Q6,5	Q7,5	Q7,6
			I0,0	I1,0	I1,1	I2,1	I2,2	I3,2	I3,3	I4,3	I4,4	I5,4	I5,5	I6,5	I6,6	I7,6	r
	MSB		Q0,0	Q1,0	Q1,1	Q2,1	Q2,2	Q3,2	Q3,3	Q4,3	Q4,4	Q5,4	Q5,5	Q6,5	Q6,6	Q7,6	r

Figure 13. CPRI Data Format (4x link - 15 bit DL)

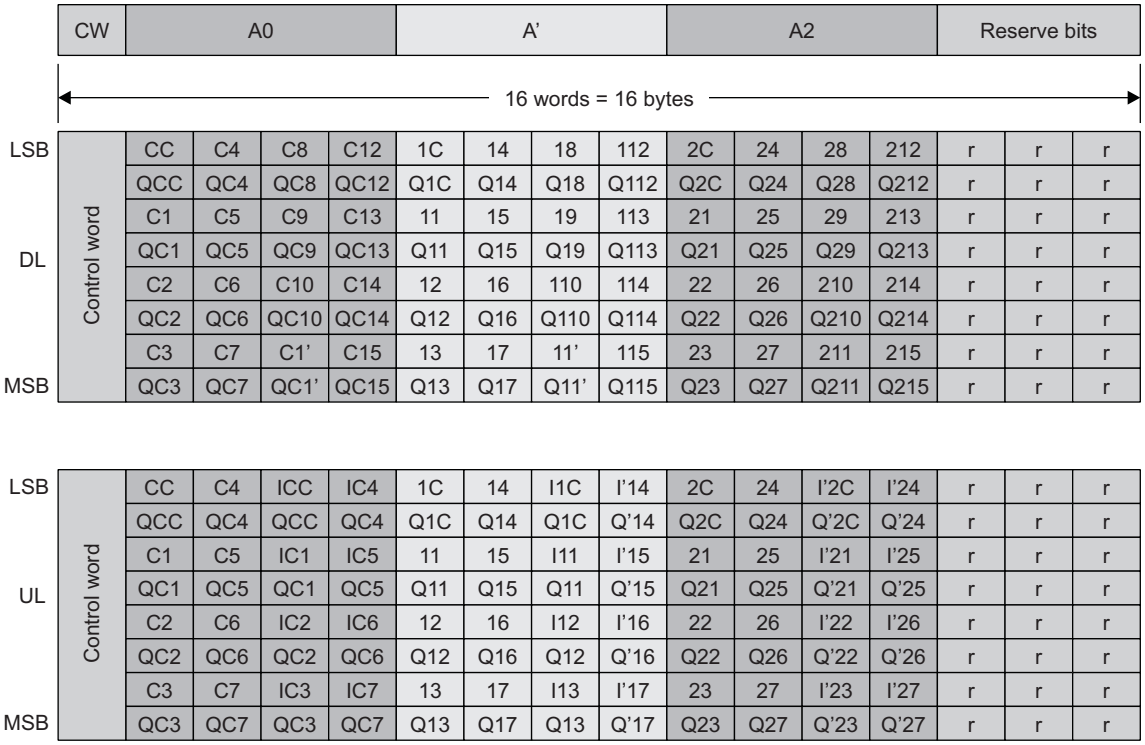


		CW	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
		16 words = 64 bytes																
	LSB	Control word	I0,0	I1,1	I2,2	I3,3	I4,4	I5,5	I6,6	I7,7								I14,14
			Q0,0	Q1,1	Q2,2	Q3,3	Q4,4	Q5,5	Q6,6	Q7,7								Q14,14
			I0,1	I1,2	I2,3	I3,4	I4,5	I5,6	I6,7	I7,8								I15,0
			Q0,1	Q1,2	Q2,3	Q3,4	Q4,5	Q5,6	Q6,7	Q7,8								Q15,0
			I0,2	I1,3	I2,4	I3,5	I4,6	I5,7	I6,8	I7,9								I15,1
			Q0,2	Q1,3	Q2,4	Q3,5	Q4,6	Q5,7	Q6,8	Q7,9								Q15,1
			I0,3	I1,4	I2,5	I3,6	I4,7	I5,8	I6,9	I7,10								I15,2
	MSB		Q0,3	Q1,4	Q2,5	Q3,6	Q4,7	Q5,8	Q6,9	Q7,10								Q15,2
	LSB	Control word	I0,4	I1,5	I2,6	I3,7	I4,8	I5,9	I6,10	I7,11								I15,3
			Q0,4	Q1,5	Q2,6	Q3,7	Q4,8	Q5,9	Q6,10	Q7,11								Q15,3
			I0,5	I1,6	I2,7	I3,8	I4,9	I5,10	I6,11	I7,12								I15,4
			Q0,5	Q1,6	Q2,7	Q3,8	Q4,9	Q5,10	Q6,11	Q7,12								Q15,4
			I0,6	I1,7	I2,8	I3,9	I4,10	I5,11	I6,12	I7,13								I15,5
			Q0,6	Q1,7	Q2,8	Q3,9	Q4,10	Q5,11	Q6,12	Q7,13								Q15,5
			I0,7	I1,8	I2,9	I3,10	I4,11	I5,12	I6,13	I7,14								I15,6
	MSB		Q0,7	Q1,8	Q2,9	Q3,10	Q4,11	Q5,12	Q6,13	Q7,14								Q15,6
DL	LSB	Control word	I0,8	I1,9	I2,10	I3,11	I4,12	I5,13	I6,14	I8,0								I15,7
			Q0,8	Q1,9	Q2,10	Q3,11	Q4,12	Q5,13	Q6,14	Q8,0								Q15,7
			I0,9	I1,10	I2,11	I3,12	I4,13	I5,14	I7,0	I8,1								I15,8
			Q0,9	Q1,10	Q2,11	Q3,12	Q4,13	Q5,14	Q7,0	Q8,1								Q15,8
			I0,10	I1,11	I2,12	I3,13	I4,14	I6,0	I7,1	I8,2								I15,9
			Q0,10	Q1,11	Q2,12	Q3,13	Q4,14	Q6,0	Q7,1	Q8,2								Q15,9
			I0,11	I1,12	I2,13	I3,14	I5,0	I6,1	I7,2	I8,3								I15,10
	MSB		Q0,11	Q1,12	Q2,13	Q3,14	Q5,0	Q6,1	Q7,2	Q8,3								Q15,10
	LSB	Control word	I0,12	I1,13	I2,14	I4,0	I5,1	I6,2	I7,3	I8,4								I15,11
			Q0,12	Q1,13	Q2,14	Q4,0	Q5,1	Q6,2	Q7,3	Q8,4								Q15,11
			I0,13	I1,14	I3,0	I4,1	I5,2	I6,3	I7,4	I8,5								I15,12
			Q0,13	Q1,14	Q3,0	Q4,1	Q5,2	Q6,3	Q7,4	Q8,5								Q15,12
			I0,14	I2,0	I3,1	I4,2	I5,3	I6,4	I7,5	I8,6								I15,13
			Q0,14	Q2,0	Q3,1	Q4,2	Q5,3	Q6,4	Q7,5	Q8,6								Q15,13
			I1,0	I2,1	I3,2	I4,3	I5,4	I6,5	I7,6	I8,7								I15,14
	MSB		Q1,0	Q2,1	Q3,2	Q4,3	Q5,4	Q6,5	Q7,6	Q8,7								Q15,14

Figure 14. CPRI Data Format (4x link - 7 UL)

		CW	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
																	A15
			← 16 words = 64 bytes →														
	LSB	Control word	I0,0	I1,1	I2,2	I3,3	I4,4	I5,5	I6,6	I'7,0	I'8,1	I'9,2	I'a,3	I'b,4	I'c,5	I'd,6	r
			Q0,0	Q1,1	Q2,2	Q3,3	Q4,4	Q5,5	Q6,6	Q'7,0	Q'8,1	Q'9,2	Q'a,3	Q'b,4	Q'c,5	Q'd,6	r
			I0,1	I1,2	I2,3	I3,4	I4,5	I5,6	I'6,0	I'7,1	I'8,2	I'9,3	I'a,4	I'b,5	I'c,6	r	If,0
			Q0,1	Q1,2	Q2,3	Q3,4	Q4,5	Q5,6	Q'6,0	Q'7,1	Q'8,2	Q'9,3	Q'a,4	Q'b,5	Q'c,6	r	Qf,0
			I0,2	I1,3	I2,4	I3,5	I4,6	I'5,0	I'6,1	I'7,2	I'8,3	I'9,4	I'a,5	I'b,6	r	Ie,0	If,1
			Q0,2	Q1,3	Q2,4	Q3,5	Q4,6	Q'5,0	Q'6,1	Q'7,2	Q'8,3	Q'9,4	Q'a,5	Q'b,6	r	Qe,0	Qf,1
			I0,3	I1,4	I2,5	I3,6	I'4,0	I'5,1	I'6,2	I'7,3	I'8,4	I'9,5	I'a,6	r	Id,0	Ie,1	If,2
	MSB		Q0,3	Q1,4	Q2,5	Q3,6	Q'4,0	Q'5,1	Q'6,2	Q'7,3	Q'8,4	Q'9,5	Q'a,6	r	Qd,0	Qe,1	Qf,2
	LSB	Control word	I0,4	I1,5	I2,6	I'3,0	I'4,1	I'5,2	I'6,3	I'7,4	I'8,5	I'9,6	r	Ic,0	Id,1	Ie,2	If,3
			Q0,4	Q1,5	Q2,6	Q'3,0	Q'4,1	Q'5,2	Q'6,3	Q'7,4	Q'8,5	Q'9,6	r	Qc,0	Qd,1	Qe,2	Qf,3
			I0,5	I1,6	I'2,0	I'3,1	I'4,2	I'5,3	I'6,4	I'7,5	I'8,6	r	Ib,0	Ic,1	Id,2	Ie,3	If,4
			Q0,5	Q1,6	Q'2,0	Q'3,1	Q'4,2	Q'5,3	Q'6,4	Q'7,5	Q'8,6	r	Qb,0	Qc,1	Qd,2	Qe,3	Qf,4
			I0,6	I'1,0	I'2,1	I'3,2	I'4,3	I'5,4	I'6,5	I'7,6	r	Ia,0	Ib,1	Ic,2	Id,3	Ie,4	If,5
			Q0,6	Q'1,0	Q'2,1	Q'3,2	Q'4,3	Q'5,4	Q'6,5	Q'7,6	r	Qa,0	Qb,1	Qc,2	Qd,3	Qe,4	Qf,5
			I'0,0	I'1,1	I'2,2	I'3,3	I'4,4	I'5,5	I'6,6	r	I9,0	Ia,1	Ib,2	Ic,3	Id,4	Ie,5	If,6
	MSB		Q'0,0	Q'1,1	Q'2,2	Q'3,3	Q'4,4	Q'5,5	Q'6,6	r	Q9,0	Qa,1	Qb,2	Qc,3	Qd,4	Qe,5	Qf,6
UL	LSB	Control word	I'0,1	I'1,2	I'2,3	I'3,4	I'4,5	I'5,6	r	I8,0	I9,1	Ia,2	Ib,3	Ic,4	Id,5	Ie,6	I'f,0
			Q'0,1	Q'1,2	Q'2,3	Q'3,4	Q'4,5	Q'5,6	r	Q8,0	Q9,1	Qa,2	Qb,3	Qc,4	Qd,5	Qe,6	Q'f,0
			I'0,2	I'1,3	I'2,4	I'3,5	I'4,6	r	I7,0	I8,1	I9,2	Ia,3	Ib,4	Ic,5	Id,6	I'e,0	I'f,1
			Q'0,2	Q'1,3	Q'2,4	Q'3,5	Q'4,6	r	Q7,0	Q8,1	Q9,2	Qa,3	Qb,4	Qc,5	Qd,6	Q'e,0	Q'f,1
			I'0,3	I'1,4	I'2,5	I'3,6	r	I6,0	I7,1	I8,2	I9,3	Ia,4	Ib,5	Ic,6	I'd,0	I'e,1	I'f,2
			Q'0,3	Q'1,4	Q'2,5	Q'3,6	r	Q6,0	Q7,1	Q8,2	Q9,3	Qa,4	Qb,5	Qc,6	Q'd,0	Q'e,1	Q'f,2
			I'0,4	I'1,5	I'2,6	r	I5,0	I6,1	I7,2	I8,3	I9,4	Ia,5	Ib,6	I'c,0	I'd,1	I'e,2	I'f,3
	MSB		Q'0,4	Q'1,5	Q'2,6	r	Q5,0	Q6,1	Q7,2	Q8,3	Q9,4	Qa,5	Qb,6	Q'c,0	Q'd,1	Q'e,2	Q'f,3
	LSB	Control word	I'0,5	I'1,6	r	I4,0	I5,1	I6,2	I7,3	I8,4	I9,5	Ia,6	I'b,0	I'c,1	I'd,2	I'e,3	I'f,4
			Q'0,5	Q'1,6	r	Q4,0	Q5,1	Q6,2	Q7,3	Q8,4	Q9,5	Qa,6	Q'b,0	Q'c,1	Q'd,2	Q'e,3	Q'f,4
			I'0,6	r	I3,0	I4,1	I5,2	I6,3	I7,4	I8,5	I9,6	I'a,0	I'b,1	I'c,2	I'd,3	I'e,4	I'f,5
			Q'0,6	r	Q3,0	Q4,1	Q5,2	Q6,3	Q7,4	Q8,5	Q9,6	Q'a,0	Q'b,1	Q'c,2	Q'd,3	Q'e,4	Q'f,5
			r	I2,0	I3,1	I4,2	I5,3	I6,4	I7,5	I8,6	I'9,0	I'a,1	I'b,2	I'c,3	I'd,4	I'e,5	I'f,6
			r	Q2,0	Q3,1	Q4,2	Q5,3	Q6,4	Q7,5	Q8,6	Q'9,0	Q'a,1	Q'b,2	Q'c,3	Q'd,4	Q'e,5	Q'f,6
			I1,0	I2,1	I3,2	I4,3	I5,4	I6,5	I7,6	I'8,0	I'9,1	I'a,2	I'b,3	I'c,4	I'd,5	I'e,6	r
	MSB		Q1,0	Q2,1	Q3,2	Q4,3	Q5,4	Q6,5	Q7,6	Q'8,0	Q'9,1	Q'a,2	Q'b,3	Q'c,4	Q'd,5	Q'e,6	r

Figure 15. CPRI Data Format (1x link - 16 bit DL, 8 bit UL)



		CW	A0				A'				A2				Reserve bits		
		16 words = 16 bytes															
LSB	Control word	CC	C4	C8	C12	1C	14	18	112	2C	24	28	212	r	r	r
		QCC	QC4	QC8	QC12	Q1C	Q14	Q18	Q112	Q2C	Q24	Q28	Q212	r	r	r
		C1	C5	C9	C13	11	15	19	113	21	25	29	213	r	r	r
DL		QC1	QC5	QC9	QC13	Q11	Q15	Q19	Q113	Q21	Q25	Q29	Q213	r	r	r
		C2	C6	C10	C14	12	16	110	114	22	26	210	214	r	r	r
		QC2	QC6	QC10	QC14	Q12	Q16	Q110	Q114	Q22	Q26	Q210	Q214	r	r	r
		C3	C7	C1'	C15	13	17	11'	115	23	27	211	215	r	r	r
MSB		QC3	QC7	QC1'	QC15	Q13	Q17	Q11'	Q115	Q23	Q27	Q211	Q215	r	r	r

LSB	Control word	CC	C4	ICC	IC4	1C	14	I1C	I'14	2C	24	I'2C	I'24	r	r	r
		QCC	QC4	QCC	QC4	Q1C	Q14	Q1C	Q'14	Q2C	Q24	Q'2C	Q'24	r	r	r
		C1	C5	IC1	IC5	11	15	I11	I'15	21	25	I'21	I'25	r	r	r
UL		QC1	QC5	QC1	QC5	Q11	Q15	Q11	Q'15	Q21	Q25	Q'21	Q'25	r	r	r
		C2	C6	IC2	IC6	12	16	I12	I'16	22	26	I'22	I'26	r	r	r
		QC2	QC6	QC2	QC6	Q12	Q16	Q12	Q'16	Q22	Q26	Q'22	Q'26	r	r	r
		C3	C7	IC3	IC7	13	17	I13	I'17	23	27	I'23	I'27	r	r	r
MSB		QC3	QC7	QC3	QC7	Q13	Q17	Q13	Q'17	Q23	Q27	Q'23	Q'27	r	r	r

Figure 16. CPRI Data Format (2x link - 16 bit DL, 8 bit UL)

	CW	A0		A1		A2		A3		A4		A5		A6		Reserve
		◄ 16 words = 32 bytes ►														
LSB	Control word	I0,0	I0,8	I1,0	I1,8	I2,0	I2,8	I3,0	I	I4,0	I	I5,0	I	I6,0	I	r
		Q0,0	Q0,8	Q1,0	Q1,8	Q2,0	Q2,8	Q3,0	Q	Q4,0	Q	Q5,0	Q	Q6,0	Q	r
		I0,1	I0,9	I1,1	I1,9	I2,1	I2,9	I	I	I	I	I	I	I	I	r
		Q0,1	Q0,9	Q1,1	Q1,9	Q2,1	Q2,9	Q	Q	Q	Q	Q	Q	Q	Q	r
DL		I0,2	I0,10	I1,2	I1,10	I2,2	I2,10	I	I	I	I	I	I	I	I	r
		Q0,2	Q0,10	Q1,2	Q1,10	Q2,2	Q2,10	Q	Q	Q	Q	Q	Q	Q	Q	r
		I0,3	I0,11	I1,3	I1,11	I2,3	I2,11	I	I	I	I	I	I	I	I	r
MSB		Q0,3	Q0,11	Q1,3	Q1,11	Q2,3	Q2,11	Q	Q	Q	Q	Q	Q	Q	Q	r
LSB	Control word	I0,4	I0,12	I1,4	I1,12	I2,4	I2,12	I	I	I	I	I	I	I	I	r
		Q0,4	Q0,12	Q1,4	Q1,12	Q2,4	Q2,12	Q	Q	Q	Q	Q	Q	Q	Q	r
		I0,5	I0,13	I1,5	I1,13	I2,5	I2,13	I	I	I	I	I	I	I	I	r
		Q0,5	Q0,13	Q1,5	Q1,13	Q2,5	Q2,13	Q	Q	Q	Q	Q	Q	Q	Q	r
		I0,6	I0,14	I1,6	I1,14	I2,6	I2,14	I	I	I	I	I	I	I	I	r
		Q0,6	Q0,14	Q1,6	Q1,14	Q2,6	Q2,14	Q	Q	Q	Q	Q	Q	Q	Q	r
		I0,7	I0,15	I1,7	I1,15	I2,7	I2,15	I	I3,15	I	I4,15	I	I5,15	I	I6,15	r
MSB		Q0,7	Q0,15	Q1,7	Q1,15	Q2,7	Q2,15	Q	Q3,15	Q	Q4,15	Q	Q5,15	Q	Q6,15	r

LSB	Control word	I0,0	I'0,0	I1,0	I'1,0	I2,0	I'2,0	I3,0	I'3,0	I4,0	I'4,0	I5,0	I'5,0	I6,0	I'6,0	r
		Q0,0	Q'0,0	Q1,0	Q'1,0	Q2,0	Q'2,0	Q3,0	Q'3,0	Q4,0	Q'4,0	Q5,0	Q'5,0	Q6,0	Q'6,0	r
		I0,1	I'0,1	I1,1	I'1,1	I2,1	I'2,1	I	I'	I	I'	I	I'	I	I'	r
		Q0,1	Q'0,1	Q1,1	Q'1,1	Q2,1	Q'2,1	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
UL		I0,2	I'0,2	I1,2	I'1,2	I2,2	I'2,2	I	I'	I	I'	I	I'	I	I'	r
		Q0,2	Q'0,2	Q1,2	Q'1,2	Q2,2	Q'2,2	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
		I0,3	I'0,3	I1,3	I'1,3	I2,3	I'2,3	I	I'	I	I'	I	I'	I	I'	r
MSB		Q0,3	Q'0,3	Q1,3	Q'1,3	Q2,3	Q'2,3	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
LSB	Control word	I0,4	I'0,4	I1,4	I'1,4	I2,4	I'2,4	I	I'	I	I'	I	I'	I	I'	r
		Q0,4	Q'0,4	Q1,4	Q'1,4	Q2,4	Q'2,4	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
		I0,5	I'0,5	I1,5	I'1,5	I2,5	I'2,5	I	I'	I	I'	I	I'	I	I'	r
		Q0,5	Q'0,5	Q1,5	Q'1,5	Q2,5	Q'2,5	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
		I0,6	I'0,6	I1,6	I'1,6	I2,6	I'2,6	I	I'	I	I'	I	I'	I	I'	r
		Q0,6	Q'0,6	Q1,6	Q'1,6	Q2,6	Q'2,6	Q	Q'	Q	Q'	Q	Q'	Q	Q'	r
		I0,7	I'0,7	I1,7	I'1,7	I2,7	I'2,7	I3,7	I'3,7	I4,7	I'4,7	I5,7	I'5,7	I6,7	I'6,7	r
MSB		Q0,7	Q'0,7	Q1,7	Q'1,7	Q2,7	Q'2,7	Q3,7	Q'3,7	Q4,7	Q'4,7	Q5,7	Q'5,7	Q6,7	Q'6,7	r

Figure 17. CPRI Data Format (4x link - 16 bit DL)

		CW	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
		◄ 16 words = 64 bytes ►															
LSB	Control word		00	10	20	30	40	50	60	70	80	90	100	110	120	130	140
			Q00	Q10	Q20	Q30	Q40	Q50	Q60	Q70	Q80	Q90	Q100	Q110	Q120	Q130	Q140
			01	11	21												
DL			Q01	Q11	Q21	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			02	12	22												
			Q02	Q12	Q22	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			03	13	23												
MSB			Q03	Q13	Q23	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
LSB	Control word		04	14	24												
			Q04	Q14	Q24	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			05	15	25												
			Q05	Q15	Q25	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			06	16	26												
			Q06	Q16	Q26	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			07	17	27												
MSB			Q07	Q17	Q27	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
LSB	Control word		08	18	28												
			Q08	Q18	Q28	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			09	19	29												
			Q09	Q19	Q29	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			010	110	210												
			Q010	Q110	Q210	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			011	111	211			I									
MSB			Q011	Q111	Q211	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
LSB	Control word		012	112	212												
			Q012	Q112	Q212	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			013	113	213												
			Q013	Q113	Q213	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			014	114	214												
			Q014	Q114	Q214	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			015	115	215	315	415	515	615	715	815	915	1015	1115	1215	1315	1415
MSB			Q015	Q115	Q215	Q315	Q415	Q515	Q615	Q715	Q815	Q915	Q1015	Q1115	Q1215	Q1315	Q1415

Figure 18. CPRI Data Format (4x link - 8 bit UL)

		CW	A0	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
		◄ 16 words = 64 bytes ►															
LSB	Control word		00	10	20	30	40	50	60	70	80	90	100	110	120	130	140
			Q00	Q10	Q20	Q30	Q40	Q50	Q60	Q70	Q80	Q90	Q100	Q110	Q120	Q130	Q140
			01	11	21												
UL			Q01	Q11	Q21	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			02	12	22												
			Q02	Q12	Q22	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			03	13	23												
MSB			Q03	Q13	Q23	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
LSB	Control word		04	14	24												
			Q04	Q14	Q24	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			05	15	25												
			Q05	Q15	Q25	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			06	16	26												
			Q06	Q16	Q26	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			07	17	27	37	47	57	67	77	87	97	107	117	127	137	147
MSB			Q07	Q17	Q27	Q37	Q47	Q57	Q67	Q77	Q87	Q97	Q107	Q117	Q127	Q137	Q147
LSB	Control word		I00	I10	I20	I30	I40	I50	I60	I70	I80	I90	I100	I110	I120	I130	I140
			Q00	Q10	Q20	Q30	Q40	Q50	Q60	Q70	Q80	Q90	Q100	Q110	Q120	Q130	Q140
			I01	I11	I21	I	I	I	I	I	I	I	I	I	I	I	I
			Q01	Q11	Q21	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			I02	I12	I22	I	I	I	I	I	I	I	I	I	I	I	I
			Q02	Q12	Q22	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			I03	I13	I23	I	I	I	I	I	I	I	I	I	I	I	I
MSB			Q03	Q13	Q23	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
LSB	Control word		I04	I14	I24	I	I	I	I	I	I	I	I	I	I	I	I
			Q04	Q14	Q24	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			I05	I15	I25	I	I	I	I	I	I	I	I	I	I	I	I
			Q05	Q15	Q25	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			I06	I16	I26	I	I	I	I	I	I	I	I	I	I	I	I
			Q06	Q16	Q26	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q	Q
			I07	I17	I27	I37	I47	I57	I67	I77	I87	I97	I107	I117	I127	I137	I147
MSB			Q07	Q17	Q27	Q37	Q47	Q57	Q67	Q77	Q87	Q97	Q107	Q117	Q127	Q137	Q147

Figure 19. CPRI Internal Data Format (1x link - 15 bit DL, 7 bit UL)

	CW	A0				A1				A2				A3			
		17 bytes															
MSB	Control word	Io,14	Io,7	Q0,14	Q0,7	Il,14	Il,7	Q1,14	Q1,7	I2,14	I2,7	Q2,14	Q2,7	I3,14	I3,7	Q3,14	Q3,7
		Io,14	Io,6	Q0,14	Q0,6	Il,14	Il,6	Q1,14	Q1,6	I2,14	I2,6	Q2,14	Q2,6	I3,14	I3,6	Q3,14	Q3,6
		Io,13	Io,5	Q0,13	Q0,5	Il,13	Il,5	Q1,13	Q1,5	I2,13	I2,5	Q2,13	Q2,5	I3,13	I3,5	Q3,13	Q3,5
DL		Io,12	Io,4	Q0,12	Q0,4	Il,12	Il,4	Q1,12	Q1,4	I2,12	I2,4	Q2,12	Q2,4	I3,12	I3,4	Q3,12	Q3,4
		Io,11	Io,3	Q0,11	Q0,3	Il,11	Il,3	Q1,11	Q1,3	I2,11	I2,3	Q2,11	Q2,3	I3,11	I3,3	Q3,11	Q3,3
		Io,10	Io,2	Q0,10	Q0,2	Il,10	Il,2	Q1,10	Q1,2	I2,10	I2,2	Q2,10	Q2,2	I3,10	I3,2	Q3,10	Q3,2
		Io,9	Io,1	Q0,9	Q0,1	Il,9	Il,1	Q1,9	Q1,1	I2,9	I2,1	Q2,9	Q2,1	I3,9	I3,1	Q3,9	Q3,1
LSB		Io,8	Io,0	Q0,8	Q0,0	Il,8	Il,0	Q1,8	Q1,0	I2,8	I2,0	Q2,8	Q2,0	I3,8	I3,0	Q3,8	Q3,0
MSB	Control word	Io,6	Qo,6	I'o,6	Q'o,6	Il,6	Ql,6	I'l,6	Q'l,6	I2,6	Q2,6	I'2,6	Q'2,6	I3,6	Q3,6	I'3,6	Q'3,6
		Io,6	Qo,6	I'o,6	Q'o,6	Il,6	Ql,6	I'l,6	Q'l,6	I2,6	Q2,6	I'2,6	Q'2,6	I3,6	Q3,6	I'3,6	Q'3,6
		Io,5	Qo,5	I'o,5	Q'o,5	Il,5	Ql,5	I'l,5	Q'l,5	I2,5	Q2,5	I'2,5	Q'2,5	I3,5	Q3,5	I'3,5	Q'3,5
UL		Io,4	Qo,4	I'o,4	Q'o,4	Il,4	Ql,4	I'l,4	Q'l,4	I2,4	Q2,4	I'2,4	Q'2,4	I3,4	Q3,4	I'3,4	Q'3,4
		Io,3	Qo,3	I'o,3	Q'o,3	Il,3	Ql,3	I'l,3	Q'l,3	I2,3	Q2,3	I'2,3	Q'2,3	I3,3	Q3,3	I'3,3	Q'3,3
		Io,2	Qo,2	I'o,2	Q'o,2	Il,2	Ql,2	I'l,2	Q'l,2	I2,2	Q2,2	I'2,2	Q'2,2	I3,2	Q3,2	I'3,2	Q'3,2
		Io,1	Qo,1	I'o,1	Q'o,1	Il,1	Ql,1	I'l,1	Q'l,1	I2,1	Q2,1	I'2,1	Q'2,1	I3,1	Q3,1	I'3,1	Q'3,1
LSB		Io,o	Qo,o	I'o,o	Q'o,o	Il,o	Ql,o	I'l,o	Q'l,o	I2,o	Q2,o	I'2,o	Q'2,o	I3,o	Q3,o	I'3,o	Q'3,o

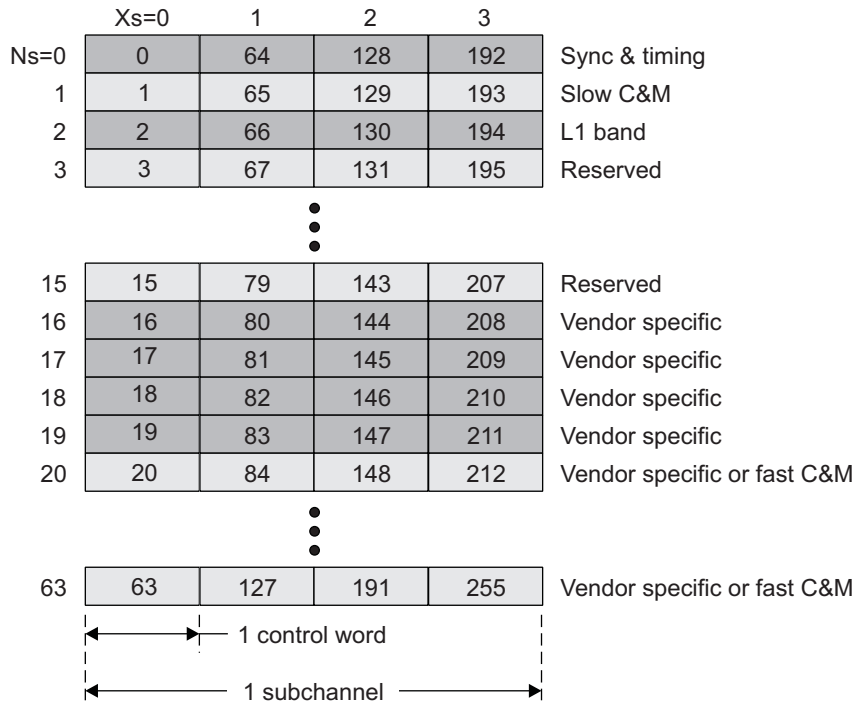
Figure 19 shows the CPRI internal data format after the RX protocol translator has unpacked the I and Q sample bits for a 1x link. The I and Q samples have been de-interleaved. The bits are packed into 16 bits (I and 16 bits Q for DL); and 8 bits I and 8 bits Q for UL. No sign extending is necessary in 8/16 bit mode; but, in 7/15 bit mode the MSB is sign extended to form either 8 or 16 bits. Figure 19 shows the 7/15 bit widths sign extended.

The 1x link internal format has room for 4 AxC streams, the 2x link internal format has room for 8 AxC streams, and the 4x link internal format has room for 16 AxC streams. If the incoming data bit width is 8/16, then the last AxC stream location is empty.

3.2.3 CPRI Control Words and Sub-Channels

The 256 control words of a hyper frame are organized into 64 sub-channels of four control words each. One hyper frame is equivalent to 256 basic frames. One sub-channel contains four control words per hyper frame, as shown in Figure 20.

Figure 20. CPRI Sub-Channels Within One Hyper Frame



For 2x and 4x links, Y>0, the sub-channel bytes for Y>0 are reserved (R). This rule applies to only sub-channel 0 (except Z.0.0) and sub-channel 2. Reserved bits are transmitted by a TX link as zeros. Reserve bits received by an RX link are not used.

The AIF transmits 256 control words every hyper frame per link. The transmitted control word information comes from one of two sources: 1) the memory- mapped registers and 2) the outbound FIFO RAMs.

Most of the memory-mapped registers can be configured at initialization and therefore do not require any software or DMA overhead during run time. Software has the option to set the BFN each frame but it is not necessary. The AIF will increment the BFN automatically. Also, the software must initialize the outbound FIFO RAMs with zeros. If this TMS320TCI6487/8 is sending the control words, then it should continuously supply the control words for every hyper frame.

Table 8. CPRI Implementation of Control Words Within One Hyper Frame

Sub-Channel Number Ns	Purpose of Sub-Channel	Xs=0	Xs=1	Xs=2	Xs=3
0	sync&timing	K28.5	HFN	BFN-low	BFN-high
1	slow C&M	slow C&M	slow C&M	slow C&M	slow C&M
2	L1 inband	version	startup	L1 inband	pointer p
3	reserved	reserved	reserved	reserved	reserved
...	...	...	...	...	...
15	reserved	reserved	reserved	reserved	reserved
16	vendor specific	vendor specific	vendor specific	vendor specific	vendor specific
17	vendor specific	vendor specific	vendor specific	vendor specific	vendor specific
18	vendor specific	vendor specific	vendor specific	vendor specific	vendor specific
19	vendor specific	vendor specific	vendor specific	vendor specific	vendor specific
...	...	...	...	...	...
p-1	vendor specific	vendor specific	vendor specific	vendor specific	vendor specific
pointer: p	fast C&M	fast C&M	fast C&M	fast C&M	fast C&M
...	...	...	...	...	...
63	fast C&M	fast C&M	fast C&M	fast C&M	fast C&M

3.2.3.1 Sub-Channel - Synchronization Data

The K28.5 control byte is inserted at Z.0.0. For 2x and 4x links, the transmitter can output either D16.2 or D5.6 on Z.0.1. The receiver accepts both outputs.

Table 9. Synchronization Control Word

CPRI line rates	#Z.0.0	#Z.0.1	#Z.0.2	#Z.0.3
1x	K28.5 (0xBC)	NA	NA	NA
2x	K28.5 (0xBC)	D16.2 (0x50) or D5.6 (0xC5)	NA	NA
4x	K28.5 (0xBC)	D16.2 (0x50) or D5.6 (0xC5)	D16.2 (0x50)	D16.2 (0x50)

The hyper-frame number is inserted at Z.64.0 and increments from 0-149. It resets to 0 at the beginning of a UMTS frame (every 10ms). The node B frame number (BFN) ranges from 0 to 4,095 and occupies 12 bits. Bits 0-7 are inserted at Z.128.0, and bits 8-11 are bits 0-3 at Z.192.0. The BFN ranges from 0 to 4,095 every 40s.

3.2.3.2 Sub-Channel - L1 Band

The CPRI protocol version byte is inserted at Z.2.0. The current version that the software must set it to is 0000 0001.

The L1 reset/alarm byte is inserted at Z.130.0, and details of the five bits are shown in Table 10. The five bits are reset, RAI, SDI, LOS, and LOF. These five bits only affect the offending link. The received alarm bits set an error/alarm bit for each condition.

The software implements the CPRI start-up state machine (described in Section 7.3.4.1) and uses the five error/alarm bits as inputs. If it is necessary to go to start-up state B; the software can force the TX link to transmit the reset, RAI, SDI, LOS, and LOF bits by setting the specific TX bits via the DMA switch fabric. Also, if the system is in start-up state B; then, the software should stop generating data for that link. Therefore, the TX link consumes all of the AIF buffered data and runs out of data. When the TX link runs out of data; it should output reserved bits for words 1-15.

Figure 21. CPRI L1 Protocol for Sub-Channel Z.130.0

31	16
Reserved	
R-0	

15	5	4	3	2	1	0
Reserved		LOF	LOS	SDI	RAI	Reset
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 10. CPRI L1 Protocol for Sub-Channel Z.130.0 Field Descriptions

Bit	Field	Value	Description
31-5	Reserved	0	Reserved
4	LOF		LOF occurs if the hyper-frame alignment cannot be achieved or is lost.
		0	No alarm
		1	Alarm
3	LOS		LOS occurs when at least 16 8b10b violations occur among a whole hyper frame.
		0	No alarm
		1	Alarm
2	SDI		SDI occurs when the link needs to shut down. SDI is determined by the software.
		0	No alarm
		1	Alarm
1	RAI		RAI occurs when either LOS or LOF is active.
		0	No alarm
		1	Alarm
0	Reset		The software can force a link to reset by setting the TX reset bit high for five hyper frames in the master. The software acknowledges reset by setting the TX reset bit high for at least ten hyper frames in the slave. The reset bit can only be sent from a master port to a slave port. The master port sets the reset request bit for at least ten hyper frames. The slave port sets the UL reset acknowledge bit for at least five hyper frames on reception.
		0	No reset
		1	Reset DL : reset request UL : reset acknowledge

The reset bit can only be sent from a master port to a slave port. The master port must set the reset request bit for at least 10 hyper frames. The slave port on reception must set the UL reset acknowledge bit for at least five hyper frames. The software can shut down a specific link by setting one of the following TX bits: reset, RAI, SDI, LOS, or LOF. To restart a link the software must deactivate the following TX bits: reset, RAI, SDI, LOS, or LOF.

The Ethernet (fast C&M) pointer p is inserted at Z.194.0. This pointer defines the sub-channel boundary between the vendor-specific region and the fast C&M region. Valid settings for the six bits of pointer p are: 0 and 20 to 63. The vendor-specific sub-channels are 16 to p-1 and the fast C&M sub-channels are p to 63. A value of 0 is not equal to any fast C&M sub-channels.

3.2.3.3 Sub-Channel - Slow C&M

The slow C&M sub-channel is defined in sub-channel 1; and, the control words are: Z.1.Y, Z.65.Y, Z.129.Y, and Z.193.Y. The data rate for the slow C&M link is defined with control word Z.66.Y. The rates are shown in Table 11.

The AIF buffers the received slow C&M data and the software later reads it. The software generates the transmitted slow C&M data and the AIF buffers it. It is transmitted later. The definition of the slow C&M data is beyond the scope of this User's Guide.

Table 11. Data Rates for Slow C&M Link

Z.66.0 bits 2to0	Description	Supported link rates	Control words	Y
0000 0000	No channel	N/A	N/A	N/A
0000 0001	240 Kbps	1x, 2x, 4x	Z.1.Y and Z.129.Y	0
0000 0010	480 Kbps	1x, 2x, 4x	Z.1.Y, Z.65.Y, Z.129.Y, Z.193.Y	0
0000 0011	960 Kbps	2x, 4x	Z.1.Y, Z.65.Y, Z.129.Y, Z.193.Y	0, 1
0000 0100	1920 Kbps	4x	Z.1.Y, Z.65.Y, Z.129.Y, Z.193.Y	0, 1, 2, 3
0000 0101…0000 0111	Invalid	N/A	N/A	N/A

3.2.3.4 Sub-Channel - Fast C&M

The fast C&M sub-channels are defined with parameter p. The possible sub-channels for the fast C&M link are 20 to 63. Table 12 shows the data rate for maximum and minimum settings of p (not including p = 0). The received fast C&M data is buffered in the AIF and the software can read it later. The software generates the transmitted fast C&M data and the AIF buffers it; it is later transmitted. The definition of the fast C&M data is beyond the scope of this User's Guide.

Table 12. Data Rates for Fast C&M Link

Line rate	# bytes per control word	Value of p	Minimum data rate (Mbps)	Value of p	Maximum data rate (Mbps)
1x	1	63	0.48	20	21.12
2x	2	63	0.96	20	42.24
4x	4	63	1.92	20	84.48

3.2.3.5 Sub-Channel – Vendor-Specific

The vendor-specific sub-channels are defined with parameter p. The possible sub-channels for the vendor-specific link are 16 to 63. Table 13 shows the data rate for the maximum and the minimum settings of p. The AIF buffers the received vendor-specific data. The software can read the data later. The software generates the transmitted slow C&M data and the AIF buffers it. It is transmitted later. The definition of the vendor-specific data is beyond the scope of this User's Guide.

Table 13. Data Rates for Vendor-Specific Link

Line rate	# bytes per control word	Value of p	Minimum data rate (Mbps)	Value of p	Maximum data rate (Mbps)
1x	1	20	1.92	0	23.04
2x	2	20	3.84	0	46.08
4x	4	20	7.68	0	92.16

3.2.3.6 Sub-Channel – Reserved

Sub-Channels Ns = 3 to 15 for Xs = 0, 1, 2, and 3 are reserved. These control words are set to zeros. The protocol encoder outputs zeros for the reserved sub-channels.

3.3 Systems that Do Not Require Either OBSAI or CPRI Protocol

AIF operates either with OBSAI or CPRI protocol. Systems that do not specifically require OBSAI or CPRI protocol to be implemented can choose any one of them.

3.3.1 Guidelines for Choosing OBSAI or CPRI Protocol:

- OBSAI offers more flexibility (but more overhead and higher link rates)
- CPRI has limited flexibility (but very little overhead and lower link rates)
- OBSAI offers combining/decombining, CPRI does not
- OBSAI enables many more options for transfer of non-IQ data (inter-TMS320TCI6487/8 communication) compared with CPRI
- OBSAI offers a max number of AxCs at a given I/Q bit width

3.3.2 Minimum Requirements for OBSAI

- The address field is required, but does not need to conform to the bit definition provided in the RP3 specification. It always is limited to ten bits.
- The specific entry for the OBSAI-type field can be any value. You can individually specify each of the 32 type values as either circuit-switched or packet-switched.
- Although OBSAI transmission rules are supported, the transmission rule implementation offers significantly more flexibility (and some limitations) for implementing transmission rules.
- The time stamp is used for reception; therefore, it must be generated properly.

3.3.3 Minimum Requirements for CPRI

- CPRI does not require any addressing or type information. The location in the data buffer is positional based.
- You do not need to use any of the CPRI control words, except for the synchronization control word (which is handled by the AIF).

4 External Implementation

4.1 Supported Topologies

The antenna external interface is implemented with six high-speed serialized links. These links support the three OBSAI RP3 and CPRI line rates. You can configure the links to operate in a daisy chain (refer to Figure 22) or a star topology (refer to Figure 23). A variation of the daisy chain (called the U-daisy chain) is also supported (refer to Figure 24). If you sue the U-daisy chain with CPRI, it may allow redundant connections. You can combine slower links into a single higher-speed link and you can extract slower-speed links from higher-speed links. Endpoint TMS320TCI6487/8 devices are defined as devices that connect directly to the RF subsystem.

Figure 22. Daisy Chain Topology

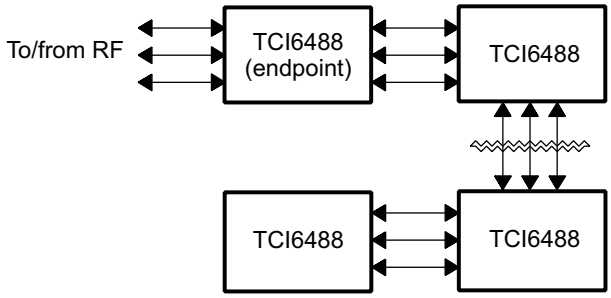


Figure 23. Star Topology

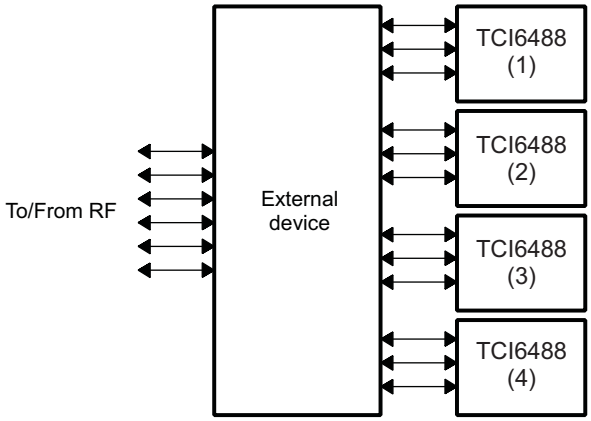
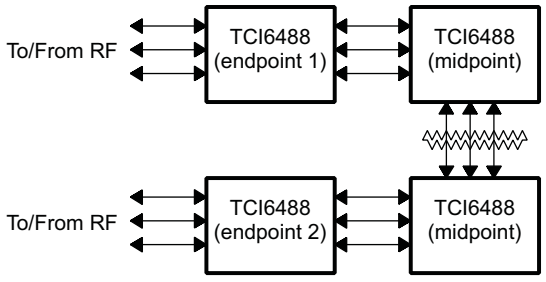


Figure 24. U-Daisy Chain Topology



4.2 Clocks

4.2.1 AIF Clocks

- Clock Inputs
  - AI_REF_CLK (the source clock for input to SERDES PLLs with variable rate, SYSCLK (N/P) clock)
  - VBUS_CLK (the main processing clock of the Antenna Interface, which is the TMS320TCI6487/8 core clock divided by 3)
- Clock Domains
  - VBUS_CLK DOMAIN (the main processing clock)
  - BYTE_CLK DOMAIN (recovered from the received SERDES data)

The primary clock inputs to the Antenna Interface are the SERDES reference clock (AI_REF_CLK) and the main processing clock (VBUS_CLK). The AI_REF_CLK is the input clock to the SERDES PLLs; there are numerous clock rates that the SERDES PLLs can accept, since the PLLs have programmable multiplier rates. The VBUS_CLK is the same frequency as the AIF VBUS interfaces.

The SERDES RX recovered clocks can become unstable if the link is interrupted (for example, during a board hot-swap). Under these conditions, the clock and local circuits that are connected to the clock may have erroneous operation. Asynchronous sideband signals are double resynchronized into the VBUS_CLK domain and they are used to isolate this erroneous behavior to the recovered clock domain.

The main processing clock is the VBUS_CLK which nominally operates at approximately 333 MHz.

Table 14. Minimum VBUS_CLK Rates vs. BYTE_CLK Rates

Max Link Rate	OBSAI RP3 BYTE_CLK rate	CPRI BYTE_CLK rate	Min VBUS_CLK rate
4x	307.2 MHz	245.76 MHz	333 MHz
2x	153.6 MHz	122.88 MHz	200 MHz
1x	76.8 MHz	61.44 MHz	200 MHz

The frame sync signal (received from the frame sync module) is asynchronous to the VBUS_CLK domain. This signal is double resynchronized prior to its use, preventing any meta-stability issues.

Table 15. Valid Input OBSAI RP3 Clock Rates vs. PLL Multiply Factors

Input Clock, MHz	PLL Multiply Factor
61.44	25
76.8	20
102.4	15
122.88	12.5
128	12
153.6	10

Table 16. Valid Input CPRI Clock Rates vs. PLL Multiply Factors

Input Clock, MHz	PLL Multiply Factor (1)
61.44	20
81.92	15
102.4	12
122.88	10
153.6	8

(1) Although all of these are valid multipliers you should follow the recommended multipliers listed in the TMS320TCI6487/8 Hardware Design Guidelines document.

The frame sync signal (received from the frame sync module) is asynchronous to the VBUS_CLK domain. This signal is double-resynchronized prior to its use, preventing any meta-stability issues.

4.2.2 Frame Sync Module Input Options

The frame synchronization interface marks the boundaries of UMTS frames and/or system time so that events can be generated which are synchronized with this time.

There are three sets of synchronization inputs to the frame synchronization module:

- The first set is the FRAME_BURST and FSYNC_CLOCK pair. This pair represents the OBSAI RP1 (*TMS320TCI6487/8 Frame Synchronization User Guide*) compliant interface.
- The second set is the ALTFSYNCPULSE and ALTFSYNCCLK pair. This pair is used as an alternative to the RP1 interface for synchronization to UMTS time. The Antenna Interface depends on using this pair for non-OBSAI synchronization mode.
- The third set is the TRT and TRT_CLOCK pair. The TRT synchronization interface is needed because the antenna interface driven by the RP3 timer always needs to run at a derivative of a UMTS clock rate. There may be non-UMTS standards that run at a different clock/sync rate, such as TRT and TRT_CLOCK. This pair is also used as an alternative to the RP1 interface for synchronization to system time.

The AIF timer drives the frame sync and the DMA event timings for the AIF. Therefore the clock source must be derived from the same clock source as the AIF SERDES reference clock. The System timer is intended to generate events for software processing and requires no timing relationship with AIF. For this reason, only the AIF timer is directly relevant to AIF. The input options for AIF timer and system timer in frame sync module are shown in the Table 17:

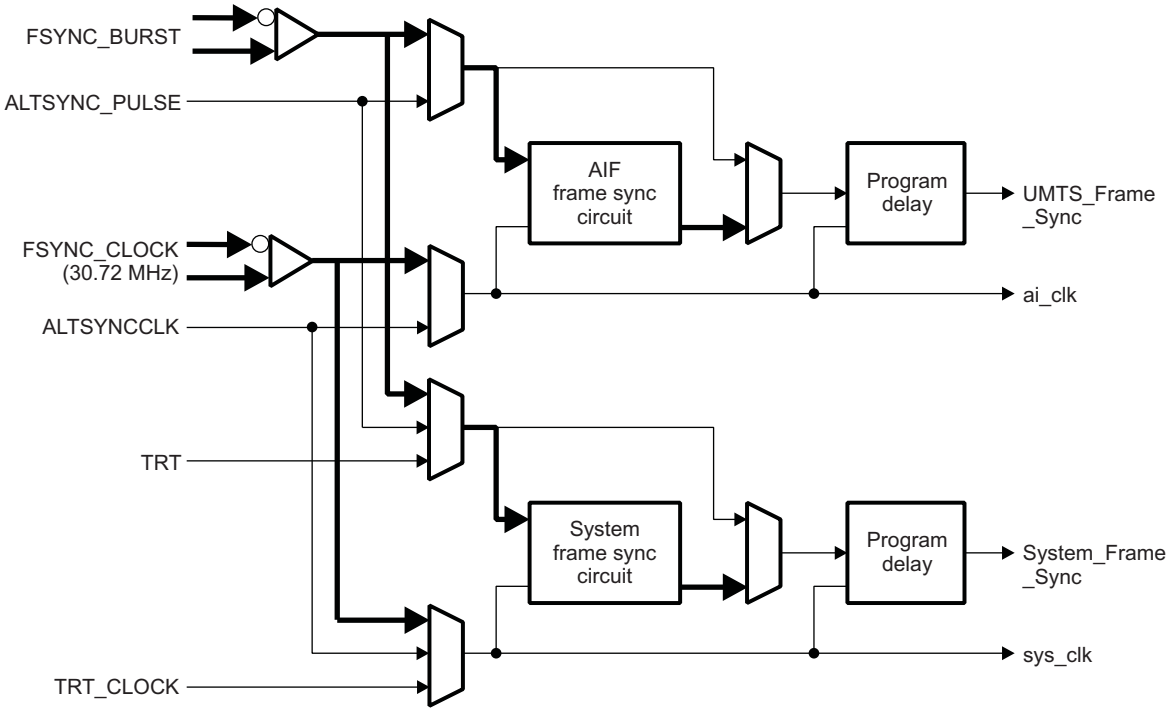
Table 17. Frame Sync Module Input Options

AIF Timer Sync	AIF Timer Clock	System Timer Sync	System Timer Clock	Intended Use
FRAME_BURST	FSYNC_CLOCK	FRAME_BURST	FSYNC_CLOCK	RP1 or non-RP1 differential sync, differential clock
ALTFSYNCPULSE	ALTFSYNCCLK	ALTFSYNCPULSE	ALTFSYNCCLK	RP1 or non-RP1 single-ended sync, single-ended clock
ALTFSYNCPULSE	FSYNC_CLOCK	ALTFSYNCPULSE	FSYNC_CLOCK	RP1 or non-RP1single-ended sync, single-ended clock
ALTFSYNCPULSE	ALTFSYNCCLK	TRT_SYNC	TRT_CLOCK	Non-RP1 or non-UMTS single-ended sync, single-ended clock

Note: If FSYNC_CLOCK is used for the AI_timer, the frequency must be 30.72 MHz and it must be derived from the same clock that provides the AIF reference clock (AI_REF_CLK or SYSCLK). If ALTFSYNCCLK is used for the AI_timer, it must be derived from the same clock as the AIF reference clock and it can be 1x, 2x, 4x, 8x, 16x or 32x of the UMTS chip rate.

Figure 25 shows the possible paths for frame synchronization.

Figure 25. RP1 Input Path



4.2.3 Frame Sync Clock

There are four clock inputs to the frame synchronization module. They are: FSYNC_CLK, VBUS_CLK, ALTFSYNCCLK, and TRT_CLK.

Clock selection is described in Table 18 and Table 19. Synchronization input is selected per Table 18.

Table 18. AIF Clock Selection

Ctrl reg 4:2 input select	Test_ mode	AI_CLK	Comment
Xx	1	VBUS_CLK	DFT (manufacturing test mode), functional mode (default after reset)
00	0	VBUS_CLK/3	Functional test mode
01	0	TRT_CLK	Non-RP1, single-ended input
10	0	FSYNC_CLK	OBSAI RP1 or non-RP1, differential input
11	0	ALTFSYNCCLK	OBSAI RP1 or non-RP1, single-ended input

Table 19. System Timer Clock Selection

Ctrl Reg 7:5 input select	Test_ mode	SYS_CLK	Comment
Xx	1	VBUS_CLK	DFT (manufacturing test mode), Functional mode (default after reset)
00	0	VBUS_CLK/3	Functional test mode
01	0	TRT_CLK	Non-RP1, single-ended input
10	0	FSYNC_CLK	OBSAI RP1 or non-RP1, differential input
11	0	ALTFSYNCCLK	OBSAI RP1 or non-RP1, single-ended input

4.2.4 Frame Sync Signal

4.2.4.1 Non-RP1 Sync

For interfaces other than OBSAI RP1, the TMS320TCI6487/8 must synchronize to the external UMTS frame or standard-specific alignment. You have the option to choose to use either differential signals {FSYNC_BURST, FSYNC_CLOCK} or single-ended input signals {ALTFSYNCPULSE, ALTFSYNCCLK} for the AIF timer. The system timer selects between {ALTFSYNCPULSE, ALTFSYNCCLK}, and {TRT, TRT_CLOCK} for the non-OBSAI supported standards. The alternate frame synchronization mechanism simply latches in a positive pulse using the clock input. This clock can be chip rate x1, x2, x4, x8, x16, or x32. . The clock for the system timer is completely up to you.

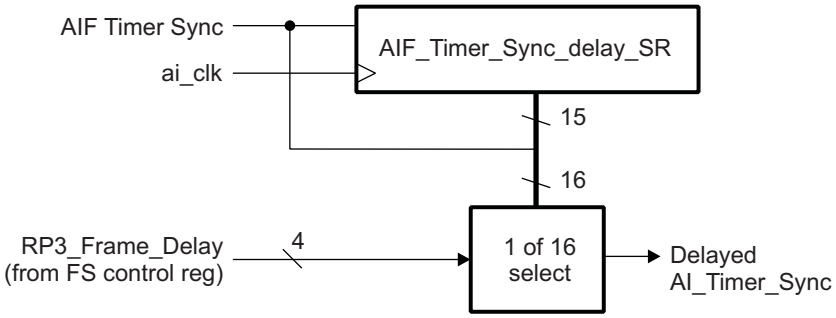
The minimum length of the UMTS_SYNC or TRT is one clock period for their associated clock.

4.2.4.2 Sync Select and Offset

Sync select muxes the AIF timer sync versus the system timer sync signals. The system timer sync signal is expected to be a minimum of one input clock period wide, high true. There is a rising edge detect circuit and an offset generator that can offset the selected sync signal 0 to 15 input clocks after the sync select mux.

The offsets are determined by the numbers that are programmed into the control register RP3_Frame_Delay, System_Frame_Delay, and TOD_Frame_Delay fields. TOD offset is only used when using the RP1 interface. The start of frame can be offset by up to 15 selected clocks. The mechanism used for the RP3 frame sync shown in Figure 26.

Figure 26. RP3 Frame Sync Offset



The same mechanism is used for TOD and system synchronizations, but use their respective Frame_delay from the frame sync control register, their respective early synchronizations and clocks.

Synchronization input is selected as outlined in Table 20.

Table 20. Sync Selection

Ctrl reg 1:0 input select	RP3_sync	Sys_sync	Comment
00	FRAME_BURST	FRAME_BURST	OBSAI RP1 mode, differential input (default on reset)
01	ALTSYNCPULSE	TRT	Non-RP1, single-ended input
10	ALTSYNCPULSE	ALTSYNCPULSE	OBSAI RP1 or non-RP1, single-ended input
11	Test_sync	Test_sync	Functional test mode

Note: If the control register RP3=system bit is set, only the system timer is used by the event generators.

4.2.4.3 RP1 Sync

For OBSAI RP1 support, a CCM (clock and control module) provides a 30.72 MHz system clock and periodically sends synchronization bursts to the baseband modules. The frame sync module receives these synchronization bursts where UMTS timing information is extracted. The RP3 and system frame numbers, frame boundary timing, and time of day information is passed in the synchronization bursts.

The OBSAI synchronization burst transmits serially over a single differential input (differential signaling) which is clocked in by way of the differential SCLK. (Same as TMS320TCI6487/8 differential FSYNC_CLOCK input shown in Figure 26.) Each bit of the serial transfer is held for eight system clock (SCLK) periods (approximately 260ns). For fields with more than a single bit, the least significant bit (LSB) is sent first.

The first field is the start bit, which marks the beginning of the synchronization burst. The eight-bit type field follows and identifies the type of information that is contained in the synchronization burst payload. The 64-bit information or payload field contains the relevant data (either frame number or time of day). The CRC field is used for data integrity and the end field terminates the synchronization burst packet. The mechanism used for the RP1 frame sync shown in Figure 27.

Figure 27. Synchronization Burst Format

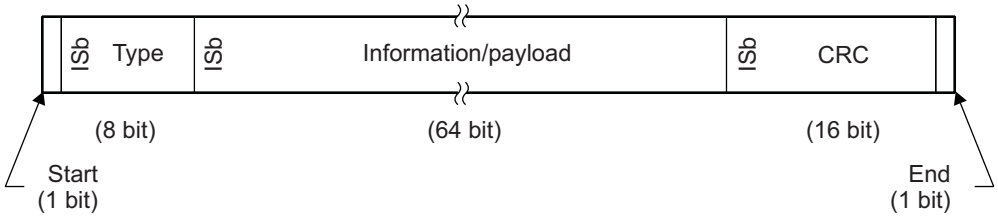


Table 21. Type Field Definition

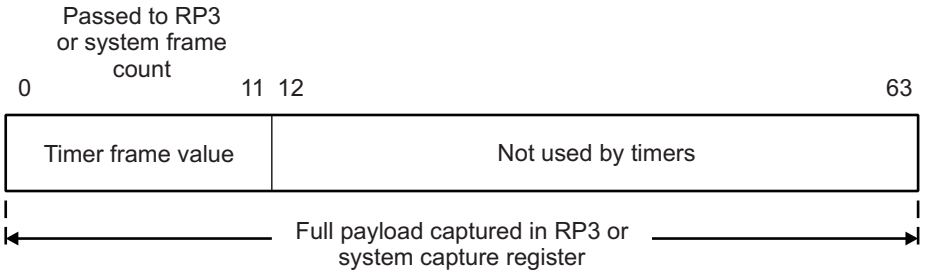
Type	Supported	Value
Not Used[1]	N/A	00h
RP3 Bus (FDD) Frame Number	Yes	01h
WCDMA/FDD Frame Number	Yes	02h
GSM/Edge1 Frame Number	No	03h
GSM/Edge2 Frame Number	No	04h
GSM/Edge3 Frame Number	No	05h
WCDMA/TDD Frame Number	No	06h
CDMA2000 Frame Number	Yes	07h
Time of Day	Yes	08h
Reserved[1]	N/A	09h – 7Fh
Spare[2]	No	80h – FFh

(1) Reserved and Not Used Types cause an Alarm/Error condition and the timer is not started.
(2) Spare Types are disregarded and the timer is not started.

The frame synchronization module supports the RP1 interface type WCDMA/FDD.

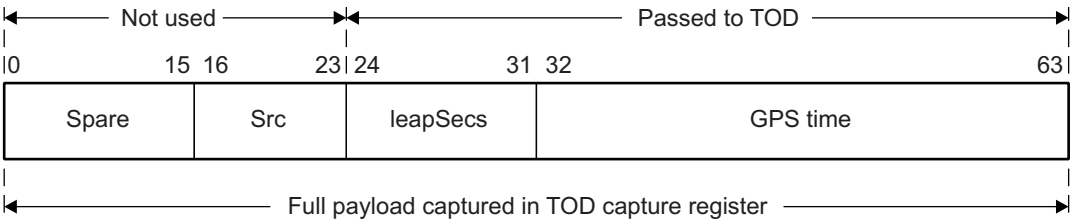
The payload is used as shown in Figure 28 for the RP3 and system timers.

Figure 28. Payload Use for Type RP3 and WCDMA FDD



The payload is used as shown in Figure 29 for the time of day timer.

Figure 29. Payload Use for Type Time of Day



4.3 High Speed Serial Interface Implementation

4.3.1 Electrical specifications

- SERDES use CML signaling and should be AC-coupled
- Receiver supports adaptive active equalization
  - Superior data dependent jitter tolerance over a lossy channel
  - < 50 mVdfpp receiver input sensitivity at sample point
- Transmitter supports selectable de-emphasis settings
- Eight selectable output amplitude settings
  - Accurately controlled using an internal band gap reference

These features allow you to adjust the settings based on your particular implementation (device-to-device connection, device over backplane, etc.).

4.3.1.1 References for Electrical specifications

- When operating in OBSAI protocol, the specifications are compatible with the OBSAI electrical requirements described in the *OBSAI RP3, v3.0 specification.*
- When operating in CPRI, the SERDES interface is compliant to the *XAUI Electrical Specification (IEEE 802.3-2002), dated 2002.*

4.3.2 Layout Guidelines

- Requires 100ohm differential impedance
- Differential pair traces must be length within +/- 5mils
- Other signals must be 2x spacing
- No stubs
- Number of vias must be minimized on both sides of differential pair

4.3.2.1 Reference for Layout guidelines

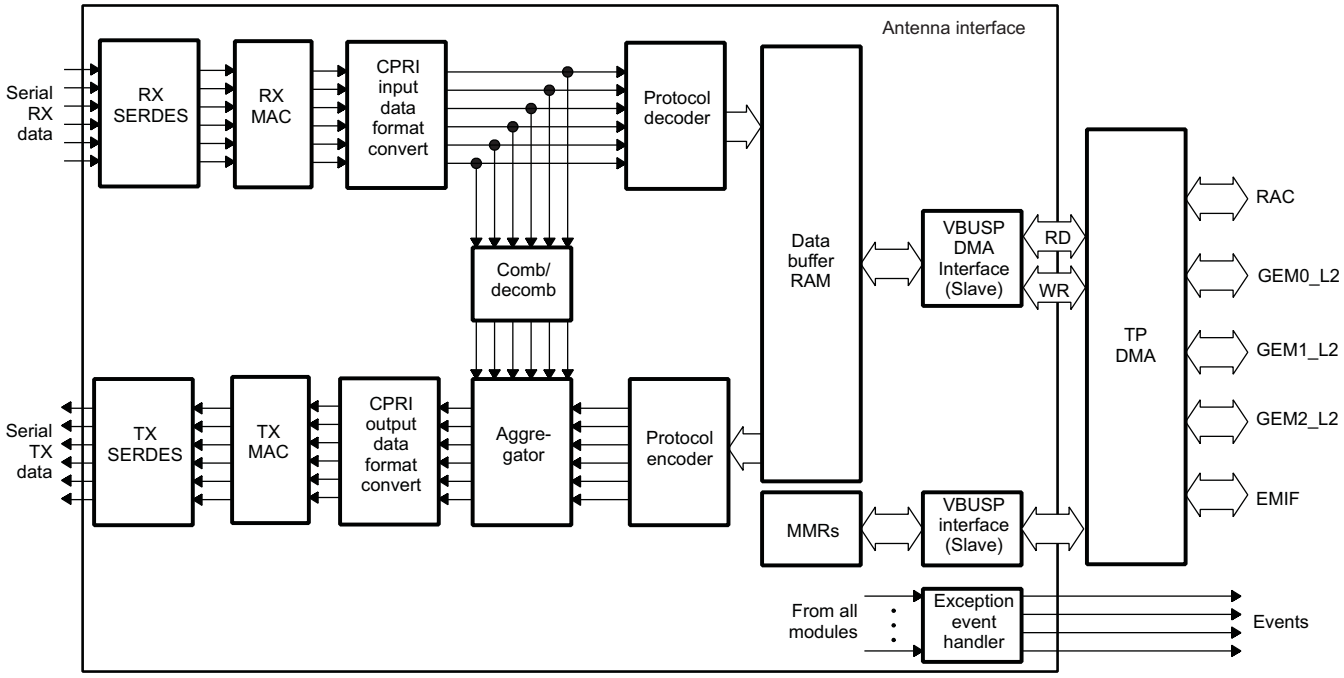
- TMS320TCI6487/8 SERDES Interface Hardware Design Guidelines

4.3.3 SERDES Configuration

- AIF is made of one quad bidi-link and 1 dual bidi-link
- Protocol Support
  - 2.0 - 4.25Gbps Serializers / Deserializers
    - Each bidi link is selectable: full, , and rates
    - Conformant to IEEE 802.3ae clause 47 (XAUI), gigabit ethernet, and fibrechannel
    - Compatible with serial rapid IO, OBSAI, and CPRI
- Flexibility
  - Eight selectable output amplitude settings
    - Accurately controlled using an internal band gap reference
  - No external loop components: single monolithic PLL design
    - Software selectable
  - 4, 5, 6, 8, 10, 12.5, 15, 20, and 25 Ref CLK input multiply ratios
    - The PLL rate should be of the full line rate
    - Each link can be configured for the full line rate, the line rate, or the line rate
- Test
  - Built in PRBS generation (verification not supported)
    - supports 2^7–1 and 2^{23}–1
    - Allows comprehensive system testing without added core logic
  - IEEE1149.6 JTAG boundary scan support
  - Internal loopback supported: TX → RX within SERDES
  - External loopback RX → TX can be supported using pass-through

5 Functional Description

Figure 30. Antenna Interface Block Diagram



5.1 Overview

The AIF module converts serial data flowing on the backplane to byte format data which is captured and stored in the data buffer RAM. The links carry circuit- switched and packet-switched data. The RAM in the AIF module contains separate inbound and outbound buffers to store circuit-switched data and packet-switched data. In addition, the AIF can perform combining, de-combining, and redirection of inbound link(s) to a specified outbound link(s). The AIF can also perform aggregation of an inbound link with an outbound link.

The functional partitioning of the antenna interface is shown in Figure 30.

AIF sub-modules:

- SERDES (six links)
- Rx MAC
- Tx MAC
- RX protocol translator
- TX protocol translator
- Combiner / decombiner
- Aggregator
- Protocol decoder
- Protocol encoder
- Data buffer RAM
- VBUSP DMA bus interface
- VBUSP configuration bus interface
- Exception event handler

The AIF module supports both OBSAI and CPRI protocols.

The SERDES, MACs, aggregator, protocol encoder, and protocol decoder support both the OBSAI RP3 protocol and the CPRI protocol. The RX protocol translator and the TX protocol translator are only used with the CPRI protocol. The combiner/decombiner is only used with the OBSAI RP3 protocol.

Note: Differences in processing OBSAI and CPRI protocols are detailed through the block descriptions.

5.2 SERDES

- There are two SERDES macros. One has four links and the other has two links.
- The term SERDES refers to both serializer and de-serializer macros together.
- Each macro has a SERDES reference clock.
- Six serial links at 1x, 2x, or 4x line rates
- 12-byte clock domains
- Receiver converts serial binary data into 10 bit blocks.
- Transmitter converts 10 bit blocks into serial stream.

Table 22. OBSAI Line Rates

LINE RATE	DATA RATE	Max # of Streams per Link
768Mbps	614.4Mbps	4
1.536Gbps	1.2288Gbps	8
3.072Gbps	2.4576Gbps	16

Table 23. CPRI Line Rates

LINE RATE	DATA RATE	Max # of AxCs 7 bit UL 15 bit DL	Max # of AxCs 8 bit UL 16 bit DL
614.4Mbps	491.52Mbps	4	3
1.2288Gbps	983.04Mbps	8	7
2.4576Gbps	1.96608Gbps	16	15

5.3 Rx MAC

- Performs 8b10b decoding
- Moves data from the BYTE_CLK domain to the VBUS_CLK DOMAIN
- OBSAI RP3 and CPRI receiver state machines
- The Rx MAC indicates the boundary of a master frame (K28.7) for OBSAI RP3 or hyper frame (K28.5 and HFN + 149) for CPRI.
- Pi measurement. (Pi offset indicates the earliest possible time instant when a Master Frame can be received. This time instant called reference time is equal to frame sync tick plus pi offset value.)
- A special character, K30.7, indicates a received data error (OBSAI only). The Rx MAC detects this and creates an error event and converts the associated byte data to 0x00 for the PD and CD. No K character indicator generates when a K30.7 character is detected.

There is also CPRI specific functionality as follows. Along with finding the K28.5 Hyper-frame boundary, the Rx MAC captures the following control bytes:

- HFN (Z.64.0)
- BFN (Z.128.0 and Z.192.0)
- Protocol Version (Z.2.0)
- Start-up (Z.66.0)
- L1 LSAR (Z.130.0)
- Pointer P (Z.194.0)

This information is available as status. It may also trigger an error / alarm event.

- In the case of OBSAI RP3, the frame boundary will occur one clock cycle before the data is ready for CD and PD. In the case of CPRI, the frame boundary will occur at the same clock cycle when the data is available.

5.4 Tx MAC

- Performs 8b10b encoding
- Moves data from the VBUS_CLK domain to the BYTE_CLK domain
- OBSAI RP3 and CPRI transmitter state machine
- The Tx MAC creates the frame structure based on the programmed link rate of 1x, 2x or 4x speed links (SD_TM_LINK_RATE). This includes the insertion of K28.7 for OBSAI RP3 (master frame boundary) and K28.5 IDLE bytes (message group boundary). It also includes the insertion of K28.5 comma bytes for CPRI for a 1x rate link, plus D16.2 and D5.6 bytes for 2x and 4x rate links. The first byte of the Master frame is transferred at an offset delta from the frame sync tick.
- If the data alignment between the incoming data stream and the TxMAC does not match, empty messages (in the form of all data bytes = 0xFF) are transmitted in the case of OBSAI; or, zeros (NULL bytes) are inserted in the case of CPRI.

5.5 RX Protocol Translator

- De-interleaves the alternating IQ data bits
- Reverses the bit order
- Only used in CPRI protocol

5.6 TX Protocol Translator

- Interleaves the alternating IQ data bits
- Reverses the bit order
- Only used in CPRI protocol

5.7 Combiner

- Combines from one to four slower links into a single faster link
- Creates an empty link for DSP created links
- Re-directs same speed links to other links
- You can only perform combining with endpoint TMS320TCI6487/8s using the OBSAI RP3 protocol
- The combine function is only used in the OBSAI RP3 protocol
- See Section 7.1.7 for CSL functions

5.8 Decombiner

- Decombines a single faster link into one to four slower links
- Decombining can only be performed with endpoint TMS320TCI6487/8s using the OBSAI RP3 protocol
- Only used in OBSAI RP3 protocol

5.9 Aggregator

- Sums IQ data together from different sources, if needed
- The IQ data can have 7/15 bit or 8/16 bit precision
- Saturates resulting summation to 16/8 bits or 15/7 bits
- Passes control and other data types that do not need aggregation.
- Supports insertion of PE messages in empty links from the combiner/decombiner, thus creating new links that you can use for inter-device communication data.
- Message insertion exists in the case of OBSAI, but not in the case of CPRI. However, AG has the capability to insert or pass thru CPRI control words and AxC container data on a per-byte basis from either the CD or the PE

5.10 Protocol Decoder

- Extract antenna carrier messages from byte stream
- Stores the payload data in the data buffer RAM
- Calculates the data buffer RAM address from the message header information (OBSAI) or from the message location (CPRI)
- Extracts control packets from the byte stream and directs them to the appropriate data buffer FIFO destinations
- Inbound memory destinations:
  - OBSAI
    - 3x packet-switched FIFOs
    - 1x packet-capture error FIFO
    - Circular circuit-switched data buffers
  - CPRI
    - Control word data buffers
    - Circular circuit-switched data buffers
  - CPRI does not provide any address or type fields as in OBSAI. The antenna carrier is inferred from byte position within the payload and the antenna carrier is used to reference the CPRI LUT for CPRI.
  - The OBSAI format provides a K 28.7 character once per UMTS frame and CPRI provides a K 28.5 character once per CPRI hyper-frame. The PD OBSAI/CPRI FSM conveniently wraps at these frame boundaries. The PD does not specifically look for these K characters, it only determines if a K character is present.
  - The expected time stamp counter increments every four chips for OBSAI or every chip for CPRI.
  - Time stamp checking is only performed on OBSAI circuit-switched data. CPRI mode and OBSAI packet-switched data do not use time stamps.

5.11 Protocol Encoder

- Synchronizes messages and frame construction with byte stream
- Reads appropriate payload data from data buffer RAM
- Appends a header to the payload data
- Supports insertion of PE messages into the correct message slots for the aggregator, using transmission rules
- Reads control packets from the data buffer FIFOs and inserts them into the appropriate message slot for transmission
- Insertion of IDLE and K characters
- For simultaneous read transactions to the AIF outbound RAMs, the protocol encoder has priority over VBUSM
- Delta synchronization transmission function
- Message handling in PE
  - OBSAI:
    - Pulls payload data from DB circuit-switched buffer
    - Pulls packet-switched data from packet-switched FIFO
    - Pulls OBSAI address and type from LUT
    - Assembles OBSAI messages, constructs header from LUT address and type, FSM constructed time_stamp and circuit_switched payload data
    - Packet-switched OBSAI messages are entirely constructed from the data in packet-switched FIFO.
  - CPRI:
    - Pulls payload data from DB circuit-switched buffer
    - Pulls CPRI control words from DB packet-switched buffer
    - Assembles CPRI basic frame

– The packet-switched FIFOs are disabled and the associated packet-switched RAM is used as CPRI control word circuit-switched RAM for CPRI.
– CPRI FSM decode is implemented with hard-coded logic, as opposed to the OBSAI programmable LUT approach. CPRI has less variants than OBSAI

Note: While the PE appends the 3-byte header to the payload data for circuit-switched data, the DSP supplies the entire 19-byte message for packet-switched data.

5.12 Data Buffer RAM

- Stores the IQ data
- Stores the control data

5.13 VBUSP DMA Bus Interface

- Separate read-only and write-only 128-bit interfaces between the AIF and DMA switch fabric
- Transfers the data with a high data rate

5.14 VBUSP Configuration Bus Interface

- 32-bit interface between the AIF and configuration switch fabric
- Transfers the low data rate data, such as configuration and status information.

5.15 Exception Event Handler

- Aggregates a large number of errors/alarms from AIF to multiple system events used as DSP interrupts and triggers for data tracing
- Four Events are made available for the rest of the system.

6 Data Transfer Over the Interfaces

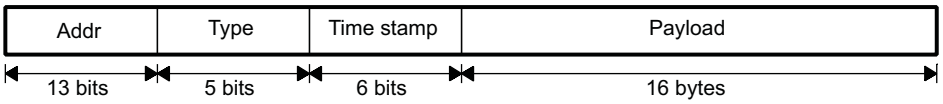
There is an external and internal data interface for the AIF. The external interface uses the SERDES links and the internal interface uses the VBUS interface. The data transferred over the two interfaces is of two basic types: circuit-switched data and packet-switched data.

6.1 Circuit-Switched Data Transfers

6.1.1 External Circuit-Switched Data Transfers via the SERDES

The external AIF interface outputs six serial streams for each of its six SERDES links. When the transmitter links are synchronized with the OBSAI RP3 transmission protocol, each link outputs messages, as shown in Figure 31 (the idle byte is treated separately). Each individual byte of the 19-byte message is 8b10b encoded from eight bits to ten bits and output is a serial SERDES stream.

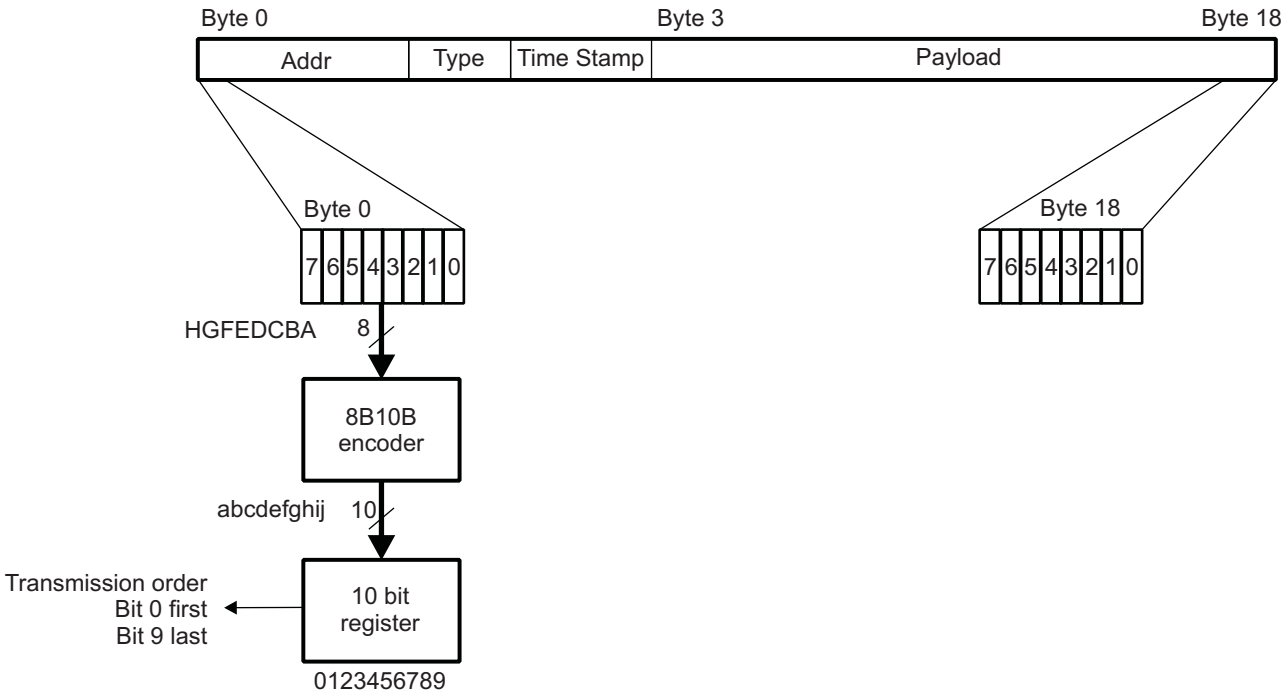
Figure 31. Byte Data Prior to 8b10b Encoding



First, the left-most byte of the address field is transmitted over the link while the right-most byte of the payload is then sent to the link. The bytes are 8b10b encoded into ten bits in the Tx MAC, and then transmitted over the SERDES link.

Figure 32 shows the order.

Figure 32. SERDES Link Transmission Order



6.1.2 Internal Circuit-Switched Data Transfers via the VBUS

The AIF internal data flows through the DMA switch fabric. The AIF VBUS is a slave device and the VBUS master is located in the DMA switch fabric. All data transfers across VBUSM are 128 bits (16 bytes) at a time. Typically, the 16 bytes are the payload data from Figure 31. You can transmit other types of data (such as control messages) through the internal interface.

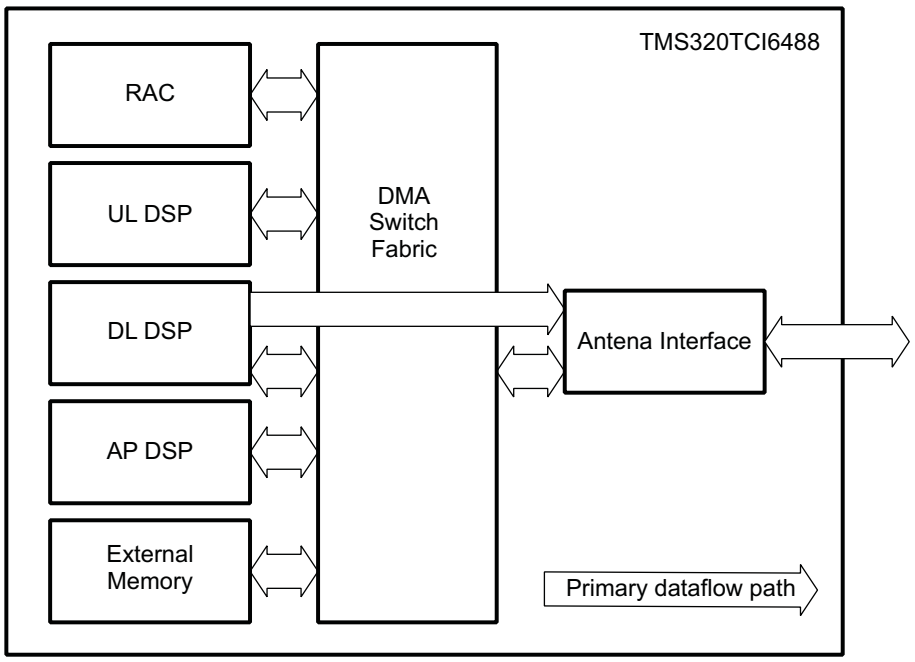
6.1.2.1 Downlink Circuit-Switched Internal Data Transfers

The downlink internal data flows are shown in Figure 33.

These flows include the following:

- Primary flow of DL DSP to antenna interface (downlink IQ data)
- A secondary flow consists of all three DSP cores generating DL data and aggregating the results on a single DSP core prior to sending the data to the AIF.

Figure 33. Downlink Internal Data Flows Via the VBUS



Downlink data consists of up to 48 streams of IQ data. Each sample is a 16-bit 2's complement number for both I and Q components.

IQ data is stored in the AIF data buffer RAM as four chips per stream. The DMA length is programmable, depending on the number of DL streams.

The TPDMA, in conjunction with the DSP cores, is responsible for creating the IQ data organization in the AIF data buffer RAM. The protocol encoder is responsible for processing and transferring the data from the AIF data buffer RAM to the aggregator. The protocol encoder also adds the header (OBSAI) or control word (CPRI) to the downlink IQ data payload to form the 19-byte message for OBSAI or 16 words for CPRI. For OBSAI the header address is the destination of the message which is typically an RF unit. The header type is typically a WCDMA/FDD type and the header time stamp is a function of the chip number within the frame. For CPRI the control word is formed by pulling the control words from DB packet-switched buffer.

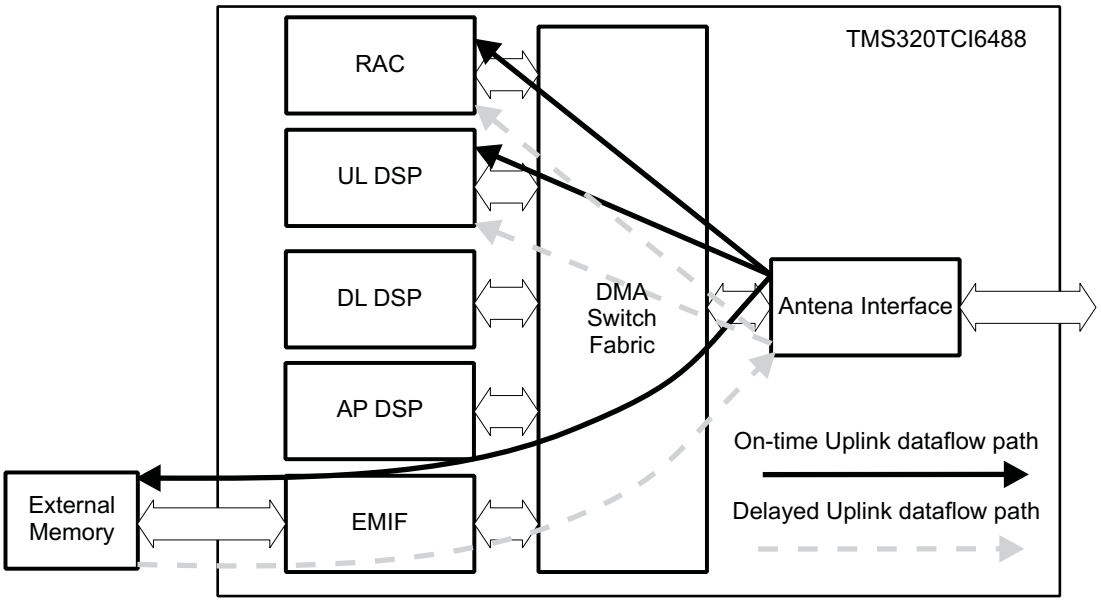
6.1.2.2 Uplink Circuit-Switched Internal Data Transfers

The uplink internal data flows are shown in Figure 34.

These flows include the following:

- Primary flow of AIF to RAC unit (Uplink IQ data)
- Primary flow of AIF to UL DSP (Uplink IQ data)
- Primary flow of AIF to external memory (delayed uplink IQ data - write)
- Primary flow of external memory to AIF (delayed uplink IQ data - read)
- Secondary flow of AIF to RAC, UL DSP, or AP DSP (use of one or more of the DSP cores RSA for uplink, control messages)

Figure 34. Uplink Internal Data Flows Via the VBUS



Uplink data from the AIF consists of up to 48 streams of IQ data. The 48 streams can be any combination of on-time and/or delayed streams. The uplink data is two times over-sampled and consists of both an even and odd sample. Each sample is an eight-bit 2's complement number for both the I and the Q components.

The protocol decoder receives messages based on address and type and strips off the header prior to storing the data into the AIF data buffer RAM. The protocol decoder is also responsible for creating the data organization. The TPDMA (DMA switch fabric) is responsible for transferring data from the AIF to its destination.

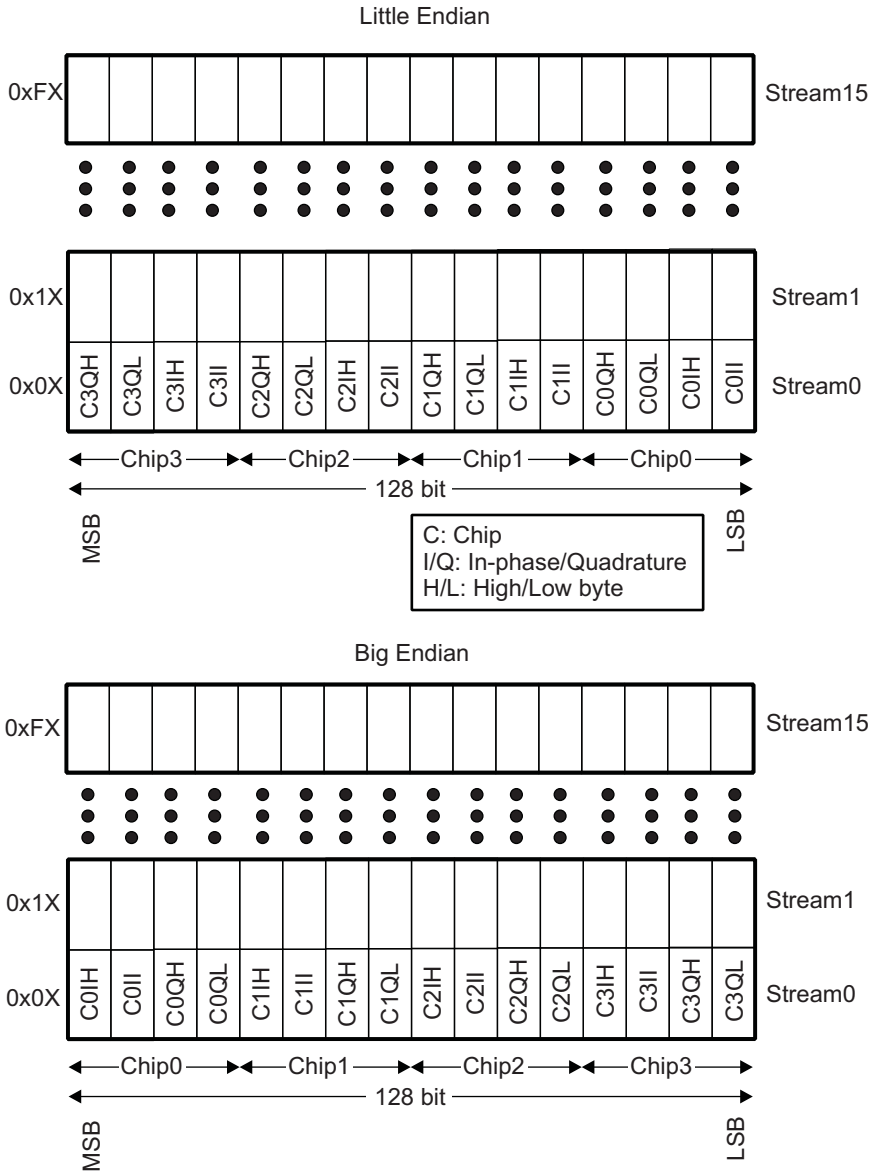
You can store the on-time uplink data in the external memory and use it later as delayed uplink data. After a defined amount of time, this data is read and you can use it in one of the two ways: 1) transfer this data directly to either the RAC or the UL DSP via the DMA switch fabric or 2) transfer the data to the AIF outbound data buffer RAM. The delayed UL data is to other TMS320TCI6487/8s and is also re-read by the DMA switch fabric and sent to either the RAC or UL DSP. The delayed UL data that is sent to other TMS320TCI6487/8 had no OBSAI RP3 header while it was stored in the external memory. The protocol encoder appends a header prior to transmission to the other TMS320TCI6487/8 devices.

6.1.2.3 Internal Circuit-Switched Data Formats

The different circuit-switched formats that are handled over the VBUS are shown below. The circuit-switched RAM is 128 bits wide, and samples are packed into these 128 bit quadwords. For all little-Endian data formats, the earliest data is packed into the least significant position and the latest data packed into the most significant position. I values are always stored next to associated Q values and the I values always occupy the least significant locations.

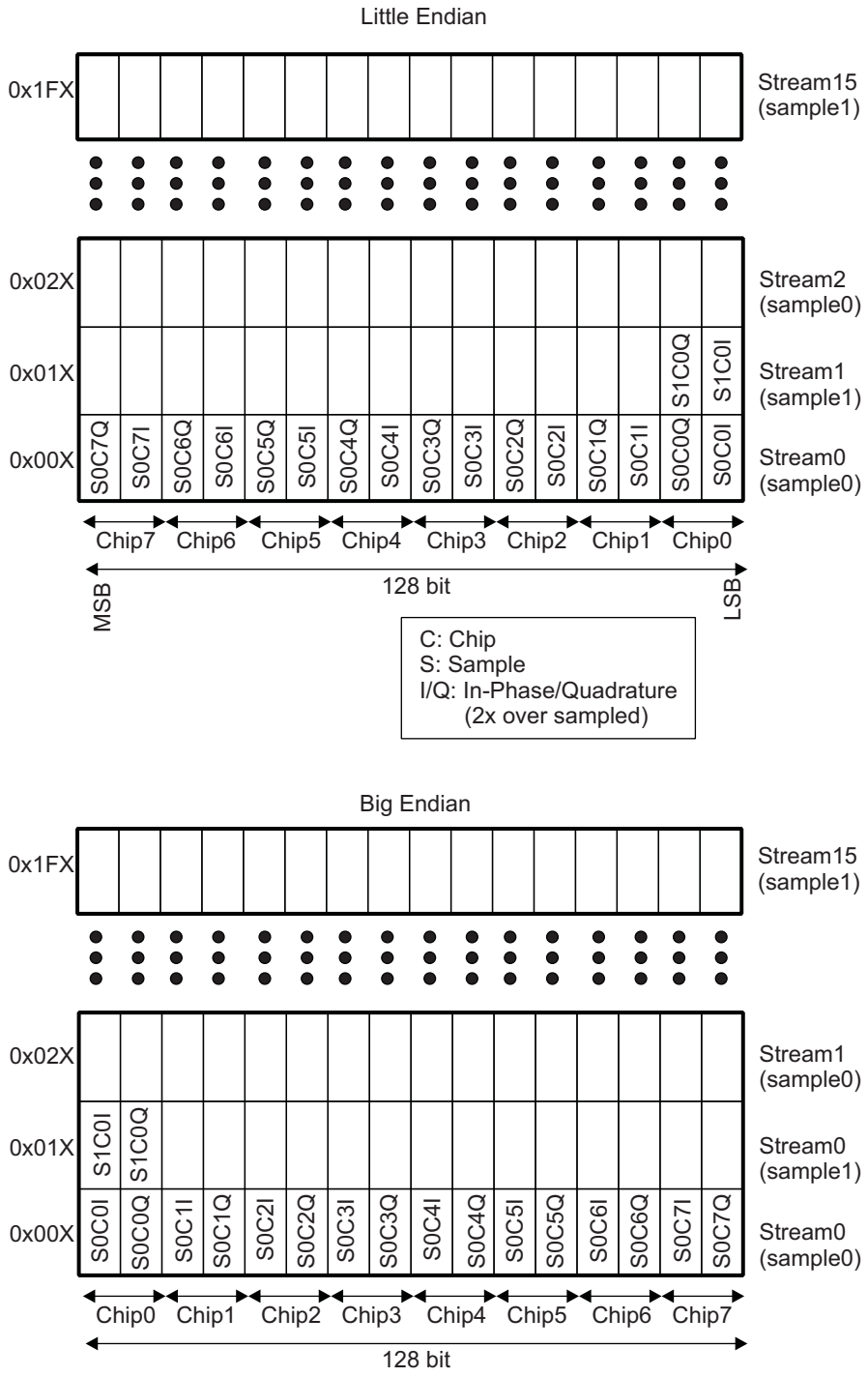
Downlink Data is stored in the circuit-switched RAM in 4-chip bursts. For a given link, these 128-bit quadwords are stored sequentially; each sequential 128-bit quadword represents a different stream/AxC. Each sample is 32 bits, 16 bit I and 16 bit Q. The I is stored in the least significant 2 bytes while Q is stored in the higher two bytes.

Figure 35. DB RAM DL Data Format



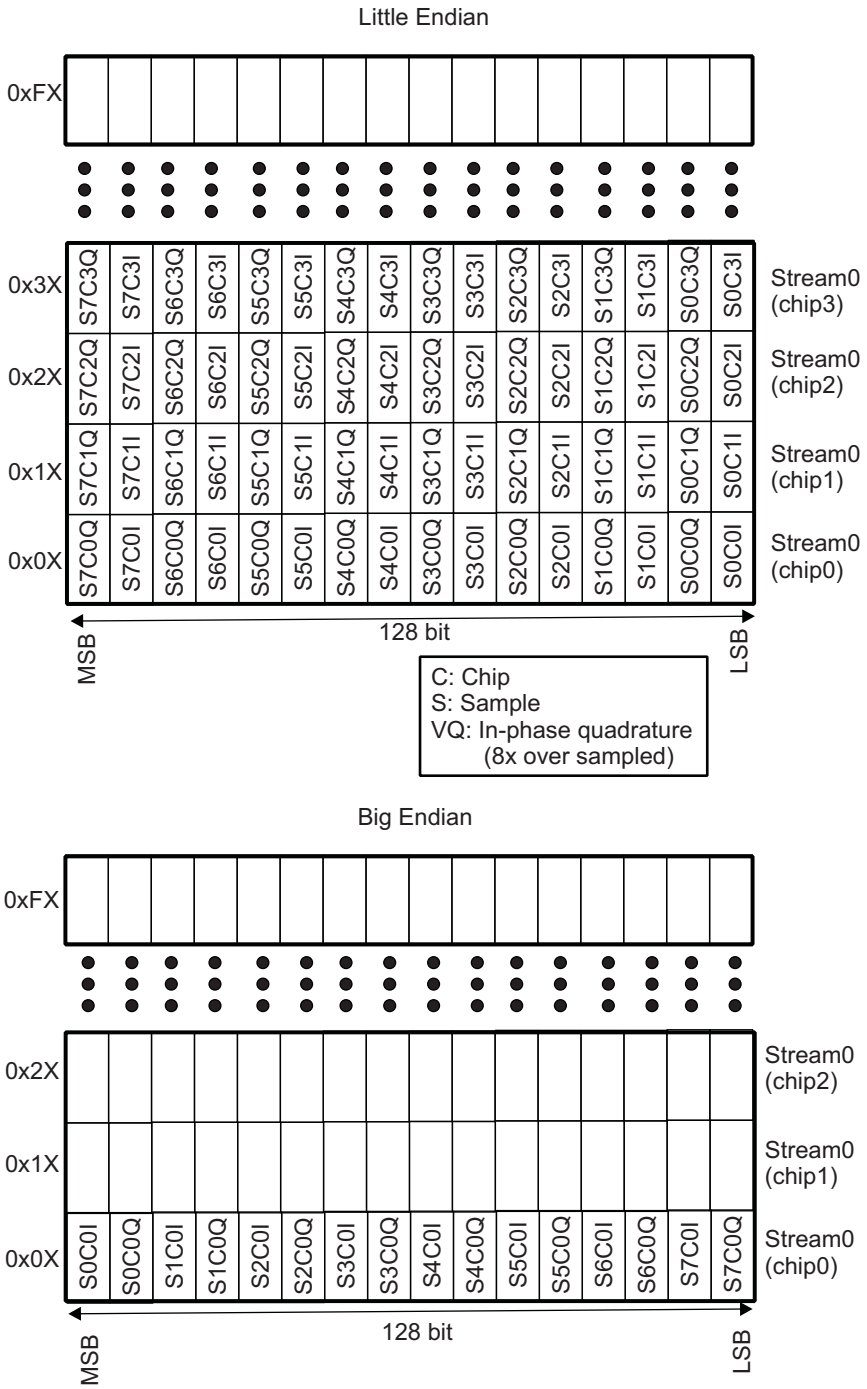
Uplink (RSA) Format data is stored in the circuit-switched RAM in 8-chip bursts. Two sequential 128 bit quadwords are used to store the 8 chips. Each sample is 16 bits, 8 bit I and 8 bit Q. UL is over sampled by a factor of 2x. Over sampled pairs, 0, 1, 0, 1, 0, 1, … are separated into different 128 bit quadwords. The first 128 bit quadword of the pair contains the even samples, and the second word constrains the odd samples.

Figure 36. DB RAM UL RSA Data Format



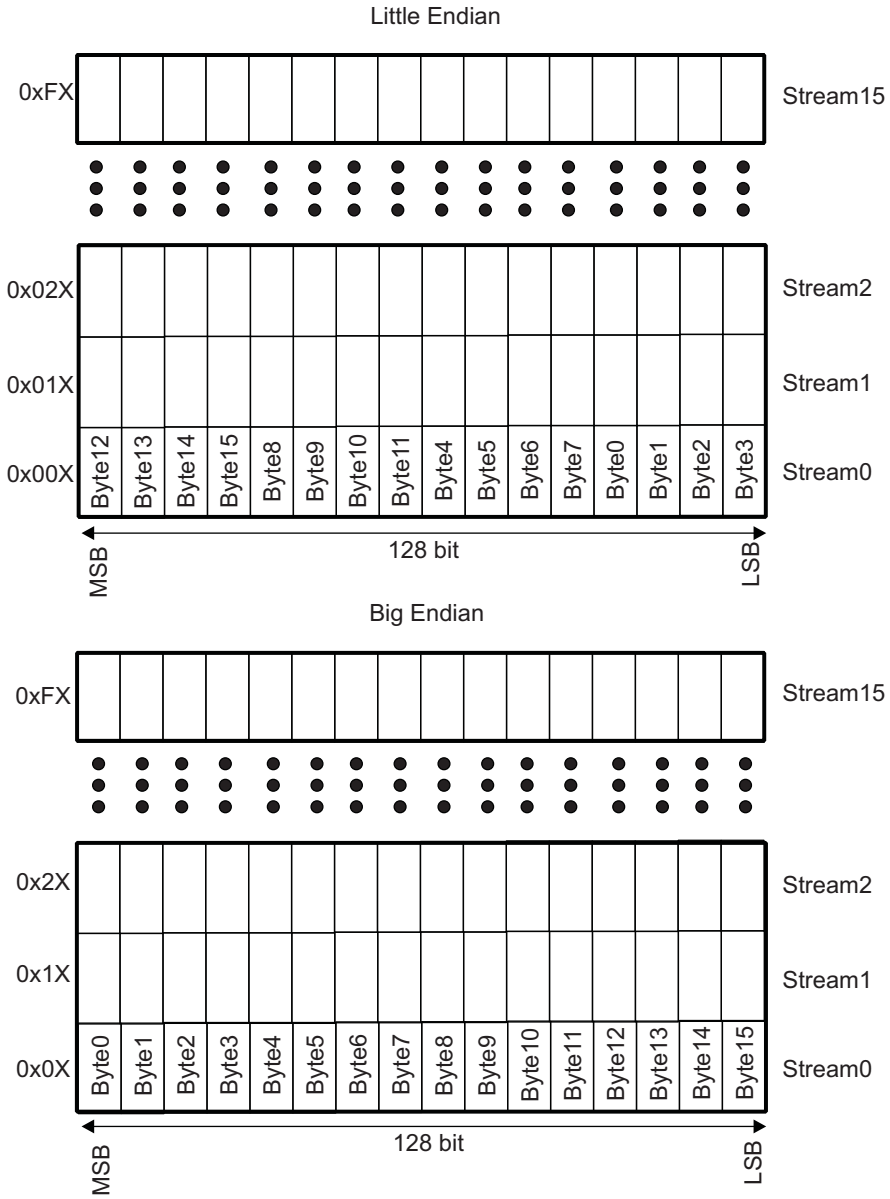
PIC MAI streams have some similarities to the other formats, but also some very unique characteristics. Similar to UL, PIC MAI samples are 8 bits I and 8 bits Q. Similar to DL, PIC MAI data is bursts of 4x chips. PIC MAI data has the unique property that it is 8x over sampled and therefore has 4x the band width per chip of either UL or DL streams. The antenna interface deals with this higher PIC MAI bandwidth by treating these streams as four separate normal bandwidth streams (Pseudo-Streams). PIC MAI streams are stored in each memory location as 8x samples for a single chip. Four consecutive memory locations store the burst of 4x chips, (Eight consecutive memory locations store a burst of 8x chips) Each of these four consecutive memory locations (chips) is transmitted in a separate OBSAI RP3 message and treated as if they were four separate streams.

Figure 37. DB RAM PIC MAI Data Format



The generic format captures whole OBSAI payloads into quadword RAM locations. Each CPRI payload relating to link rate of {1x, 2x, or 4x} requires a {1, 2, or 4} quadword. The generic format has no strict required use, but is intended to be generic DSP data. For this reason, the data is 32-bit work oriented.

Figure 38. DB RAM Generic Data Format



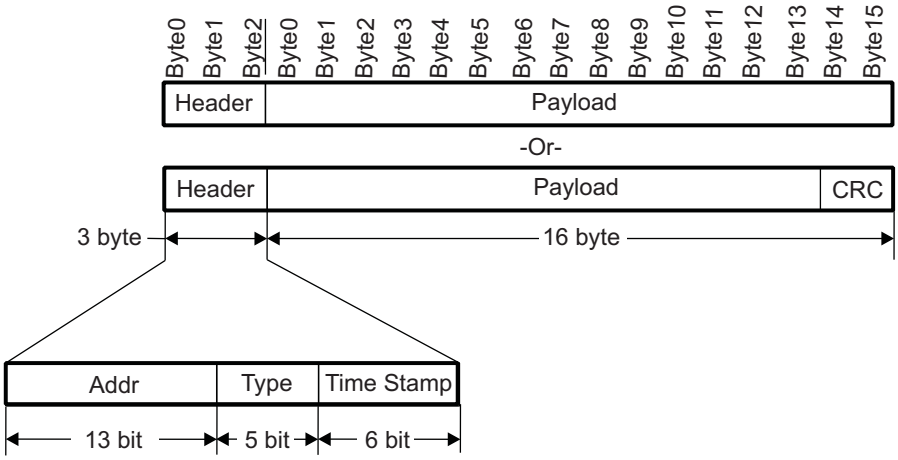
6.2 Packet-Switched Data Transfers

Note: The description given in this section regarding the packet-switched data transfers is OBSAI specific.

Packet-switched messages (control, measurement, loopback, or other programmable types using the type field) use the payload to transport a control message. The packet-switched message contains non-IQ data. The packet-switched messages are used to communicate between TMS320TCI6487/8 units and/or RF units. . Although OBSAI only defines non-IQ data for control words, TCI6487/8 allows non-IQ data to be used in data slots as well.

A higher layer protocol is used within the payload to provide error detection. The following details a control signaling protocol. As illustrated in Figure 39, the control message format consists of two payload fields: the message data and the CRC. This format is also true for loopback and measurement messages. The CRC is generated by the source DSP and then later checked by the destination DSP. A CRC may or may not be used in packet-switched messages contained in antenna carrier time slots. In this case, the entire 16-byte field would be available for the payload.

Figure 39. Packet-Switched Message Format



The timestamp field of a control, loopback, or measurement message typically contains value 0000 0000. On reception, the protocol decoder ignores the timestamp for packet-switched data. The software provides the timestamp that the protocol encoder for transmission will insert.

6.2.1 External Packet-Switched Data Transfers via the SERDES

The left-most byte of the address field is first transmitted over the link while the right-most byte of the payload is last sent to the link. The order is shown in Figure 32.

6.2.2 Internal Packet-Switched Data Transfers via the VBUS

You can initiate packet-switched messages via the RF module or by any TMS320TCI6487/8 in a chain. Inter-TCI6487/8 messaging is supported to a limited extent. Any TCI6487/8 wishing to originate a message may do so under a strict set of rules. Each TCI6487/8 is assigned a link number and a set of transmission rules for sending messages. The DSP (typically the Applications DSP) creates the contents of a message in the form of a 19-byte packet.

A queue of control messages is built inside the AIF. The protocol encoder reads these messages and inserts them into the destination link, based on the transmission rules programmed for the device.

Packet-switched messages are inserted into the data stream if the transmission rule dictates the insertion. If another packet-switched message (generated from a different TCI6487/8) occupies the same location, it is over-written and an error condition generates. Error conditions can alert or interrupt DSP software. Packet-switched slots are exclusively reserved for a given transmission rule. Once the slot is defined for packet-switched messages, it can not be reused for any other type of message. Link configuration is only supported when a given link is down or after an AIF reset.

Packet-switched messages are received based on message type (control, measurement, loopback, or other programmable types) and address. The protocol decoder receives all messages from all links. The protocol decoder is programmed with an address to match with the incoming message's address field. Only those messages with matching address fields are extracted. The protocol decoder writes these messages into a queue of packet-switched messages in the AIF data buffer RAM. The protocol decoder will write the entire 19 bytes into the AIF data buffer Ram unlike circuit-switched data where only 16 bytes of payload data is written into the AIF data buffer RAM. This allows the software to decide how to use the address and type fields and to check the CRC if necessary. The AIF asserts an interrupt to the DSP or TPDMA, signaling the arrival of a control message.

Note: The usage of the Type fields is programmable in the Antenna Interface. A look-up table in the protocol encoder controls the AIF to treat messages as circuit-switched or packet-switched. You are expected to program this look-up table so that control, measurement, and loopback types are packet-switched.

All packet-switched messages are re-transmitted (after combining / decombining). This supports a type of broadcasting.

7 Usage Details

7.1 AIF Configuration

The configuration of AIF requires configuring each active link and enabling the same. AIF operates either in CPRI or OBSAI protocol.

Note: The example CSL codes shown in this section are based on the assumption that only one link is enabled. The link is operating at 1x link rate and four streams are enabled.

7.1.1 AIF CSL Description

7.1.1.1 AIF Driver Initialization

The CSL 3.x specification requires that an initialization routine be used to initialize any variables stored within its context. The AIF driver layer does not store context information therefore, no AIF CSL initialization is required.

However, the AIF CSL needs to provide a CSL_aifInit() routine (see section 0) in order to satisfy CSL 3.x requirements. This function does not perform any processing, it merely returns a result code indicating a successful operation. Examples demonstrating AIF module usage will always call the CSL_aifInit() routine before any operation on AIF is performed to maintain consistency with CSL 3.x peripheral usage.

7.1.1.2 AIF Driver Call Sequence

Following successful initialization of the AIF module via CSL_aifInit(), the application can configure a link. The application is required to call the CSL_aifOpen() routine (see section 0) to open a handle to the desired link. Once a handle to a link is assigned it can be used to configure, monitor the status and change certain properties of the link. The AIF CSL requires links to be setup and/or modified independently i.e. properties of a link can be changed only using its own link handle. During runtime, the number of active link handles is the same as the number of active links.

Configuration of the link is performed used CSL_aifHwSetup(). Any control information (includes enabling, disabling, modifying certain properties of the link or AIF hardware) can be passed to the AIF module using CSL_aifHwControl() routine. The status of the link or AIF module can be queried using CSL_aifHwStatus(). See Section 7.1.1.4 for details on the CSL functions.

The sequence of calls is shown below:

Example 1. AIF Driver Call Sequence

```
CSL_aifInit(...); // Initialize AIF CSL driver

handleAifLink = CSL_aifOpen(...);  // Open handle for configuring link

CSL_aifHwSetup(handleAifLink,...); // Configure link

CSL_aifHwControl(handleAifLink, CSL_AIF_CMD_ENABLE_LINK,...);  // Enable link

CSL_aifGetHwStatus(handleAifLink, CSL_AIF_QUERY_OK_STATUS_BIT,...);  // Get status of AIF

// Terminating a link
CSL_aifHwControl(handleAifLink, CSL_AIF_CMD_DISABLE_LINK,...);  // Disable link before closing

CSL_aifClose(handleAifLink); // Release link resource
```

7.1.1.3 Memory Allocation

During run time, the application is responsible for memory management relating to CSL usage i.e. application is responsible for creating/destroying any objects relating to configuration, status and control. The CSL may use a negligible amount of stack memory for creating local variables.

The AIF CSL requires private memory to store look-up tables. The memory required for AIF driver usage is known at compile time and is specified in the AIF CSL release notes.

The register layer header file for the AIF is provided in cslr_aif.h. The functional layer header file for the AIF is provided in csl_aif.h.

CSL_aifInit(), CSL_aifOpen(), CSL_aifHwSetup(), CSL_aifHwControl() and CSL_aifGetHwStatus() are the only APIs used in configuring the AIF.

7.1.1.4 CSL Functions

For detailed descriptions of CSL functions, see the *TMS320TCI6487/8 Antenna Interface CSL and Driver API Specification*.

Short descriptions of the CSL functions are specified below:

- **CSL_aifInit()** is an AIF-specific initialization function. This initializes the CSL data structures, and does not touch the hardware. This function should be called before using any of the CSL APIs in the AIF module.
- **CSL_aifOpen()** opens the instance of AIF requested. The open call sets up the data structures for the particular instance of AIF device. The device can be re-opened anytime after it has been normally closed if so required. The handle returned by this call is input as an essential argument for rest of the APIs used for AIF.

  AIF must be successfully initialized via CSL_AIFInit() before calling CSL_aifOpen(). Memory for the AIF object must be allocated outside this call. This object must be retained while usage of this peripheral.
- **CSL_aifHwSetup()** initializes the device registers with the appropriate values provided through the HwSetup Data structure provided in the csl_aif.h file. Both CSL_aifInit() and CSL_aifOpen() must be called successfully in that order before CSL_aifHwSetup() can be called. The user has to allocate space for and fill in the main setup structure appropriately before calling this function.
- **CSL_aifHwControl()** performs various control operations on AIF link, based on the command passed.
- **CSL_aifGetHwStatus()** accesses the value of various parameters of the AIF instance. The value returned depends on the query passed.

CSL_fsyncInit(), CSL_fsyncOpen(), CSL_fsyncHwSetup() and CSL_fsyncHwControl() are the only APIs used in configuring the Fsync when used with AIF.

CSL_edma3Init(), CSL_edma3Open(), CSL_edma3ChannelOpen(), CSL_edma3HwChannelSetupParam(), CSL_edma3HwSetup(), CSL_edma3ParamSetup(), CSL_edma3HwChannelControl(), CSL_edma3HwControl () are the some of the APIs used in configuring the EDMA when used with AIF.

AIF and EDMA operations are based on triggers from the frame sync module.

7.1.2 Link Configuration

TMS320TCI6487/8 has six serial links. Each link is programmed for parameters Section 7.1.2.1 - Section 7.1.2.5.

7.1.2.1 Type of Transaction

- Transmitter enabled
- Receiver enabled
- Loopback (both transmit and receive bits are enabled)

7.1.2.2 Link Rates

- 1x
- 2x
- 4x

7.1.2.3 Width of Antenna Data for CPRI Protocol

- CPRI protocol
  - Downlink
    - 15-bit packed
    - 16-bit packed
  - Uplink
    - 7-bit flexible position
    - 8-bit packed
- OBSAI protocol
  - Downlink - 16 bit
  - Uplink - 8 bit

  Both transmit and receive links can be programmed as either uplink or downlink.

7.1.2.4 Number of Streams for a Link

- Max of 4 AxCs for 1x link
- Max of 8 AxCs for 2x link
- Max of 16 AxCs for 4x link

Apart from the above specified parameters, the parameters for configuring each block in the AIF, including SERDES, RxMAC, TxMAC, etc. populate and are written into the memory-mapped registers via the CSL_HwSetup() function. The CSL_HwSetup() function is called for each active link. Each link is programmed by populating four structures. The parameters that apply globally to all of the links are grouped as global configuration parameters. Parameters that are common to inbound and outbound transfer for a particular link are grouped into common parameters. Parameters that are specific to inbound or outbound transfers are grouped into inbound configuration and outbound configuration, respectively.

7.1.2.5 Split Up of CSL Structures

- Link Configuration
  - Global configuration
    - Protocol type
    - Combiner decombiner configuration
    - Packet-switch input FIFO setup
- Common configuration
  - Link index
    - Link rate
    - SERDES configuration
  - Inbound configuration
    - Width of antenna data
    - Data type
    - Number of active antenna streams for this link
    - Rx MAC configuration
    - Protocol decoder configuration
  - Outbound configuration
    - Width of antenna data
    - Data type
    - Number of active antenna streams for this link
    - Tx MAC configuration
    - Protocol encoder configuration
    - Aggregator configuration

Example 2. CSL Structures Code

```
#define AIF_NUM_ACTIVE_AXS_OUTBND_LINK_0  4
#define AIF_NUM_ACTIVE_AXS_INBND_LINK_0   4
CSL_AifHandle hAifLink;
CSL_AifLinkObj AifObj
CSL_AifContext AifContext;
//AIF module specific parameters
CSL_AifParam  aifParam;
// CSL status
CSL_Status status;
// Ctrl Argument;
Uint32 ctrlArg = TRUE;
/* Setup for links */
CSL_AifLinkSetup aConfigLink;
// global config for AIF */
CSL_AifGlobalSetup gblCfg;
// Setup for common params for links
CSL_AifCommonLinkSetup aCommoncfg;
// Setup for inbound links
CSL_AifInboundLinkSetup aInboundCfg;
// Setup for outbound links
CSL_AifOutboundLinkSetup aOutboundCfg;
// Setup for Pd for inbound links
CSL_AifPdSetup aPdCfg;
// Rx Mac setup for inbound links
CSL_AifRxMacSetup aRmCfg;
// Tx Mac setup for outbound links
CSL_AifTxMacSetup aTmCfg;
// Protocol encoder setup for outbound links
CSL_AifPeSetup aPeCfg;
// Aggegator setup for outbound links
CSL_AifAggregatorSetup aAgCfg;
```

Example 2. CSL Structures Code (continued)

```
CSL_AifPdCommonSetup pdCommonCfg;
CSL_AifInboundFifoSetup psInFifoCfg;
CSL_AifSerdesSetup aSdCfg;

//Populate the link Configuration fields with pointers to the parameter structures
aConfigLink.globalSetup = &gblCfg;
aConfigLink.commonlinkSetup = &aCommoncfg;
aConfigLink.inboundlinkSetup = &aInboundCfg;
aConfigLink.outboundlinkSetup = &aOutboundCfg;

//Populate global config fields
gblCfg.linkProtocol = CSL_AIF_LINK_PROTOCOL_OBSAI;
gblCfg.pCdSetup = NULL; // don't setup for now
gblCfg.pInboundPsFifoSetup = &psInFifoCfg
//Populate common fields
aCommoncfg.linkIndex = CSL_AIF_LINK_0;
aCommoncfg.linkRate = CSL_AIF_LINK_RATE_1x;
aCommoncfg.pSerdesSetup = &aSdCfg; // unpopulated for now

//Populate inbound fields
aInboundCfg.antDataWidth = CSL_AIF_DATA_WIDTH_16_BIT;
aInboundCfg.linkDataType = CSL_AIF_LINK_DATA_TYPE_DL;
aInboundCfg.numActiveAxC = AIF_NUM_ACTIVE_AXS_INBND_LINK_0;
aInboundCfg.pRxMacSetup = &aRmCfg;
aInboundCfg.pPdSetup = &aPdCfg;

//Populate outbound fields
aOutboundCfg.antDataWidth = CSL_AIF_DATA_WIDTH_16_BIT;
aOutboundCfg.linkDataType = CSL_AIF_LINK_DATA_TYPE_DL;
aOutboundCfg.numActiveAxC = AIF_NUM_ACTIVE_AXS_OUTBND_LINK_0;
aOutboundCfg.pTxMacSetup = &aTmCfg;
aOutboundCfg.pPeSetup = &aPeCfg;
aOutboundCfg.pAggrSetup = &aAgCfg;

//Do setup for link
CSL_aifHwSetup(hAifLink, &aConfigLink);
ctrlArg = (Uint32)TRUE;

// Enable Tx/Rx (loopback mode)
CSL_aifHwControl(hAifLink, CSL_AIF_CMD_ENABLE_DISABLE_RX_LINK,(void *)&ctrlArg);
CSL_aifHwControl(hAifLink, CSL_AIF_CMD_ENABLE_DISABLE_TX_LINK,(void *)&ctrlArg);
```

7.1.3 SERDES Configuration

To configure SERDES, the following need to occur:

- The PLL multiply factor must be programmed
- The PLL must be enabled

7.1.3.1 PLL Multipliers

For valid input clock rates and multiply factors, see Table 15 and Table 16.

Note: The SERDES PLLs must be locked before enabling the transmit and receive links.

Assuming that the INP REF CLK for the SERDES is 122.88, calculate the PLL multiply factor as:

REFCLKP/NFREQ = (LINERATE ɔRATESCALE)/ MPY

Where RATE_SCALE = 0.5 for full rate
1 for half rate
2 for quarter rate

Now MPY = (LINE_RATE × RATESCALE) ÷ 122.88

Example for 4x = .05 × 3072 ÷ 122.88 = 12.5

Therefore, the multiply factor is 7.

Table 24. PLL Multiply Factors

Multiply Factor Value	Effect
0000 0000	4x
0000 0001	5x
0000 0010	6x
0000 0011	Reserved
0000 0100	8x
0000 0101	10x
0000 0110	12x
0000 0111	12.5x
0000 1000	15x
0000 1001	20x
0000 1010	25x
0000 1011	Reserved
0000 1100	Reserved
0000 1101	50x
0000 1110	60x
0000 1111	Reserved

Note: PLL MPY remains the same for all of the rates (1x, 2x and 4x)

```
mpyFactor  =  7;
ctrlArg = (Uint32)TRUE;
// Program PLL MPY factor
   CSL_aifHwControl(hAifLink[0], CSL_AIF_CMD_MOD_SD0_PLL_MUL_FACT,
(void *)& mpyFactor);

// Enable PLL
CSL_aifHwControl(hAifLink[0], CSL_AIF_CMD_ENABLE_DISABLE_SD0_PLL,
(void *)&ctrlArg);
```

7.1.3.2 Electrical Options

Amplitude, de-emphasis, and equalization parameters must be programmed for the SERDES. These values are dependent on the specific board topology. Guidelines for specific board topologies are available in the *TMS320TCI6487/8 SERDES Hardware Designs Guidelines* document.

Example:

```
aSdCfg[0].bEnableRxAlign = TRUE;
aSdCfg[0].bEnableRxLos = FALSE;
aSdCfg[0].pCommonSetup = &sdCommonCfg;
aSdCfg[0].rxCdrAlgorithm = 0;
aSdCfg[0].rxEqualizerConfig = CSL_AIF_SERDES_RX_EQ_ADAPTIVE;
aSdCfg[0].rxPairPolarity = 0;
aSdCfg[0].rxTermination = 0;
aSdCfg[0].txAmpConfig = CSL_AIF_SERDES_TX_AMP_CONFIG_750;
aSdCfg[0].txCommonMode = 0;
aSdCfg[0].txDeEmphasisConfig = CSL_AIF_SERDES_TX_DE_CONFIG_8;
aSdCfg[0].txPairPolarity = 0;
aCommoncfg[0].pSerdesSetup = &aSdCfg[0];
CSL_aifHwSetup(hAifLink[0], &aConfigLink[0]);
```

7.1.3.3 Selection of Link Rates

The rate for each link can be selected either as 1x, 2x, or 4x.

Example:

```
aCommoncfg[0].linkRate = CSL_AIF_LINK_RATE_1x;
```

7.1.4 Topology Configuration

The antenna interface supports the daisy chain, U-daisy chain, and star topologies (see Section 4.1).

The daisy chain options offer the lowest system cost since a separate device is not necessary. There is no limit to the number of TMS320TCI6487/8 units that can be connected in a chain. However, latencies through the daisy chain need to be considered; see Table 25 for latency analysis. The star topology uses a central node that adds cost, but allows more flexibility. The difference between daisy chain and U-daisy is that the U-daisy chain uses the unused links on the end device. The following give examples of daisy chain (Figure 40) and U-daisy chain (Figure 41 and Figure 42) options. The link rates 1x, 2x and 4x are denoted by 4S, 8S and 16S signifying the number of antenna carrier streams for each rate.

Figure 40. DAISY1 - DAISY3 Options

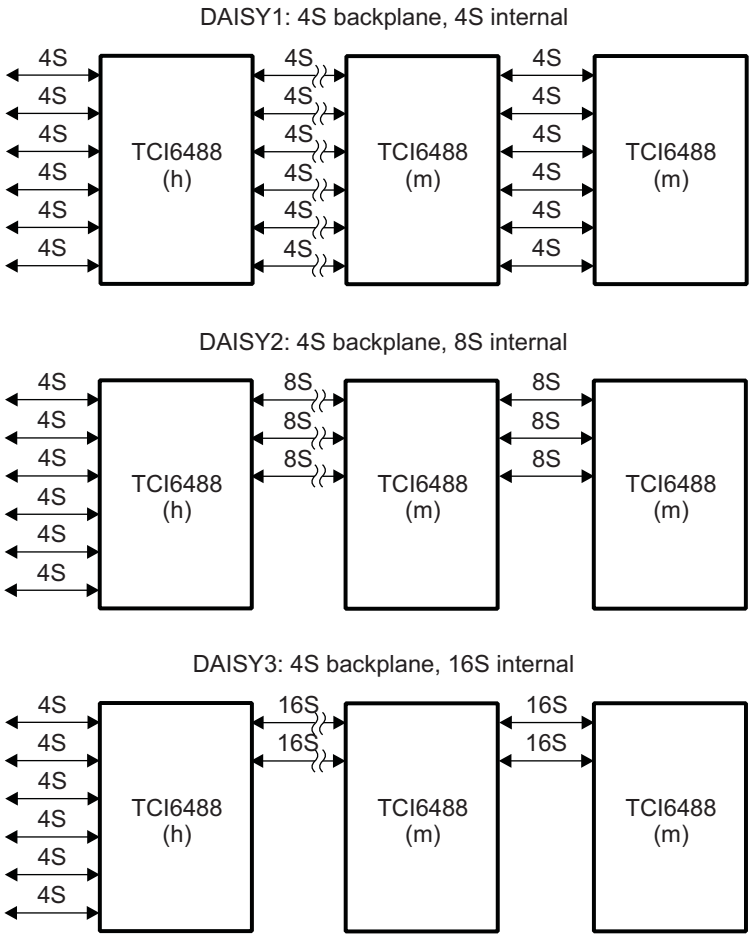


Figure 41. UDAISY1 Option, 4S Backplane Links, 16S Internal

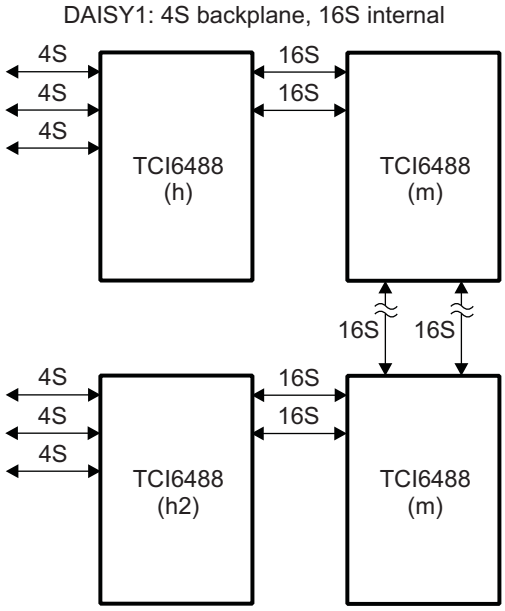
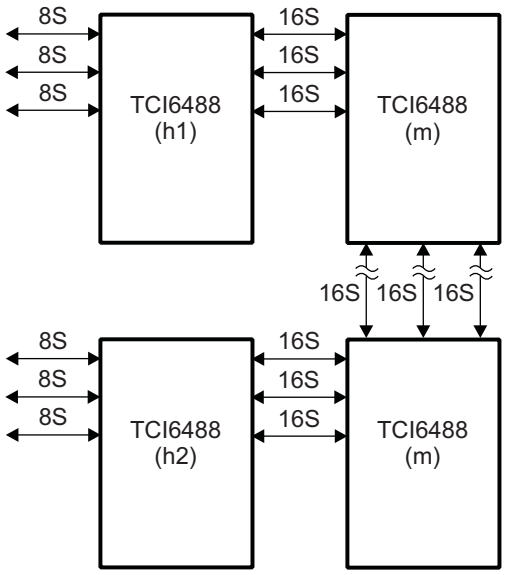


Figure 41 shows the worst case number of links (five) needed for supporting 24 uplink and 24 downlink streams using 4S links. Only 12 streams are passed over the 16S links so that not all available bandwidth is used.

In order to support 48 uplink streams, another link was needed. The additional link is shown in Figure 42.

Figure 42. UDAISY2 Option, 8S Backplane, 16S Internal



There are many other link configurations that are supported. Table 25 lists some of the other supported configurations.

For configuration of the AIF for these topologies, see Section 7.1.7.

Note: The maximum number of backplane links that a daisy chain configuration supports is four. You can also use the daisy chain with an external device if an external device is needed to add additional features or to convert the OBSAI RP3 protocol. The daisy chain topology requires fewer links to the external device, compared with a star topology and scales well (the number of links to the external device is not dependent on the number of TMS320TCI6487/8).

Table 25. Daisy Chain Link Configurations

		Backplane Links		Intermediate Links
# UL Streams	# DL Streams	TMS320TCI6487/8 (h1)	TMS320TCI6487/8 (h2)	TMS320TCI6487/8 (h1,h2,m)
12	12	3 @ 4S	Not used	3 @ 4S or 1 @ 16S
24	24	3 @ 4S	3 @ 4S	2 @ 16S
24	24	3 @ 8S	Not used	3 @ 8S
48	24	3 @ 8S	3 @ 8S	3 @ 16S
48	24	3 @ 16S	Not used	3 @ 16S
32	16	2 @ 4S, 1 @ 8S	2 @ 4S, 1 @ 8S	2 @ 16S
16	16	4 @ 4S	Not used	1 @ 16S or 2 @ 8S

7.1.4.1 General Guidelines to Configure Links for Uplink or Downlink

- RXn and TXn must be the same line rate for each value of n
- Any RX port can be mapped to any TX port
- Combining/Decombining is only supported for OBSAI
- There is a maximum of two links that can be combined/decombined into

The following lists the valid combinations:

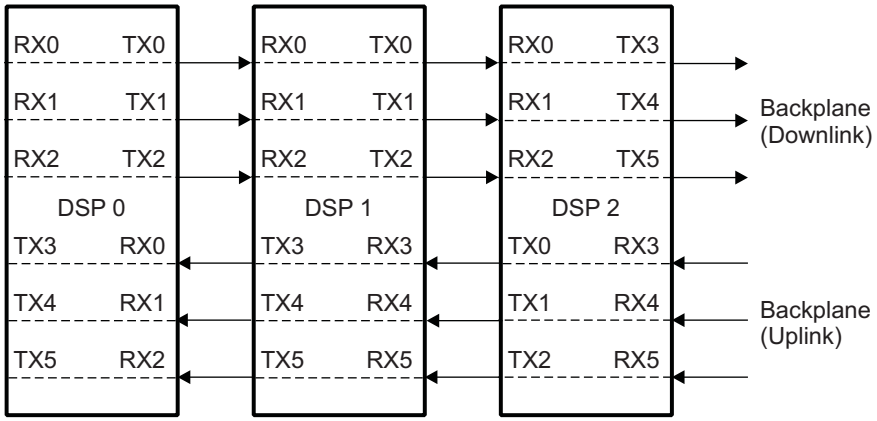
- 1 1x links are combined to 2x link
- 2 1x links are combined to 2x link
- 1 1x links are combined to 4x link
- 2 1x links are combined to 4x link
- 3 1x links are combined to 4x link
- 4 1x links are combined to 4x link
- 1 2x links are combined to 4x link
- 2 2x links are combined to 4x link
- 1 2x link and 1 1x link are combined to 4x link
- 1 2x link and 2 1x links are combined to 4x link
- Any input link can be directed to any output link at the same rate (pass through mode)
- One Input link may go to only one output link
- Only standard WCDMA/FDD OBSAI RP3 links can be combined.
- Substitute empty links (useful for inserting DSP created links) for inactive links or failed links
- Input links can only be combined in any programmable order
- Unassigned combined links can be used for inter-TMS320TCI6487/8 communication (either circuit switched or packet-switched)

7.1.4.2 Mapping of Incoming and Outgoing Links

Example Configuration #1 (no combining/decombining)

- Three 1x uplink streams and three 1x downlink streams to the backplane
- All links run at 1x
- Notice that the mapping between TX and RX is different for the DSP connected to the backplane. This is done to demonstrate the flexible mapping, but is not required (i.e., RX3-RX4 could have been connected to the backplane).

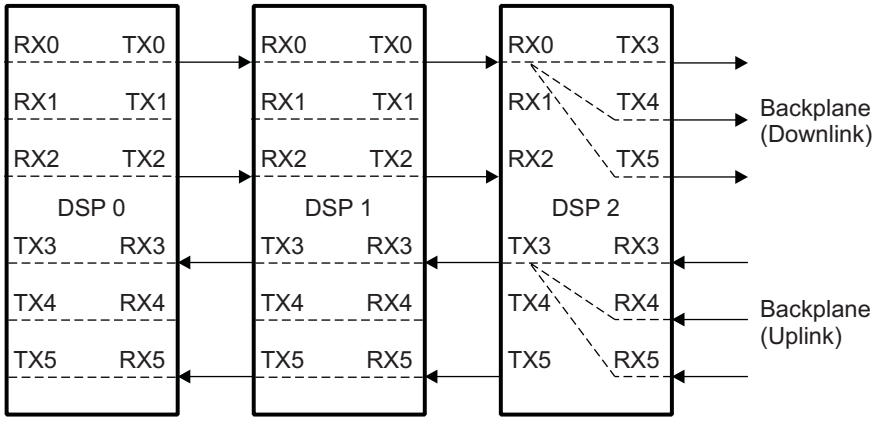
Figure 43. Configuration Without Combining and Decombining



Example Configuration #2 (with combining/decombining, OBSAI only)

- Three 1x uplink streams and three 1x downlink streams to the backplane
- Three links are combined / decombined to one 4x stream between TMS320TCI6487/8 units
- Unused links are shown being used for inter-DSP communication

Figure 44. Configuration With Combining and Decombining



7.1.5 Configure Transmit Links

For transmit links, apart from configuring the outbound parameters explained in Section 7.1.2, Tx MAC, protocol encoder and aggregator modules must be configured.

7.1.5.1 Transmission Rules Usage

Each link is controlled with two sets of look-up tables. One look-up table has 21 entries and the other table has 84 entries. These two LUTs need to be programmed in the case of OBSAI mode only.

The 21-entry look-up table has entries for 16 AxC Streams, 4 possible streams of control slots and 1 for packet-switched data in message slot.

The information in the 21-entry look-up table is transmission rule mask, transmission rule compare, header address, header type, data format, and control information for the aggregator. The possible data formats are downlink, uplink, MAI, and generic. The possible aggregator control information is whether the data is 7/8 bits, 15/16 bits, insert a message, or pass the data through the aggregator.

The information in the 84-entry look-up table is data type, index to the 21 entry table, and packet-switch data location. The data type is either circuit-switched data or packet-switched data. The packet-switched data location is used to define whether the packet-switch data is located in one of the four possible control message slots (maximum of four in four consecutive message groups) or located in a data message slot.

Five separate transmission counters are maintained for each link. One of these is dedicated to message slots and the other four are used to maintain control slots. The data slot counter counts 0-1919 each frame regardless of 4x, 2x, or 1x link rate. The control counters each represent one of the four 1x control slot sets contained within a 4x link. Each of these control counters is intended to increment at the 1x link rate. The counters are reset every 10ms regardless of the link rate.

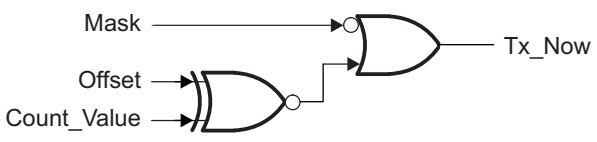
For example:

- For a 4x link, each of the four control slot counters updates exactly once per 84 count.
- For a 1x link, only one of the four control message counters is used. The 84 count will be comprised of 4 repeating counts of 21. Within each 21 count, the control slot counter will have been accessed/enabled exactly once (or 4 times per 84 count)

The protocol encoder's 84-count look-up function retrieves transmission rule counter control information. Each state of the 84-count look-up indicates a control slot or data slot and the control fields to maintain and use the transmission rule counters. These fields are as follows:

- Counter_Select:
  - Selects the transmission rule counter to use (4x control or 1x data counters)
  - If selected causes the control counter to update the next CLK_CYCLE
- Mask:
  - Control slot counter: termination count
  - Data slot counter: mask field indicating 2n modulo type periodicity
- Compare:
  - Control slot counters: the offset value is used as equal check
  - Data slot counters: offset value

Figure 45. PE Data Slot Count Use



With each count of the 84 count, fields are retrieved for use with the transmission rules and for data slots. The fields are used as a modulo 2^n mask and an offset. In determining whether or not to transmit in a given slot, the *mask* value qualifies which bits should be evaluated. Of the bits to be evaluated, the *offset* is checked for equivalence with the counter value. With this approach, the circuitry supports any 2^n modulo periodicity and any offset.

For example:

Mask = 11'b000_0000_0111 (indicates periodicity of 8)

Offset = 11'b000_0000_0010 (indicates offset of 2)

Tx_Now when

Counter = 11'b000_0000_0010

= 11'b000_0000_1010

= 11'bxxx_xxxx_x010

For control slots, the control counters are enabled and maintained. The control counters mimic the generic modulo concept quite closely. These counters count up to the modulo value minus one. The increment_count should be set dependent on the link rate: once per 21 messages for 1x link, once per 42 messages for 2x link, and once per 84 messages for 4x link. With this type of operation, any random periodicity or offset can be supported. The 84 look-up Field 0 represents the termination count of the selected control counter. Once the counter reaches its programmable termination count, the counter wraps to zero.

The Field1 Offset value for the control counters is used as an equivalency check. When the control counter value is equal to Field1, the transmission is triggered (Tx_Now).

Programming transmission rules involve programming the 84CNT_LUT (see Section 8.2.12.7), the ID_LUT (see Section 8.2.12.8 and Section 8.2.12.9), and TERM_CNT_CFG (see Section 8.2.12.5 and Section 8.2.12.6). The PE maintains a counter that is incremented for each message slot until it reaches the count of 84 after which it is reset to zero. This counter is used to index into the 84_CNT_LUT. The 84_CNT_LUT has the following fields to be programmed:

- AXC_OR_CTRL_INDEX_LUT: Index for accessing the identity LUT. Also the stream index for data slots is used in circuit switched address generation.
- CKT_VS_PKT_SWTCH_MSG_LUT: circuit switched data or packet-switched data.
- TIME_STAMP_INCR_NXT_LUT: Controls Time Stamp to increment, Incremented value is used for next message. The suggested OBSAI setting for this field is:
  - 1x - 84'h8 4444 4222 2211 1110 8888
  - 2x - 84'h8 0202 0202 0200 8080 8080
  - 4x - 84'h8 0000 8000 8000 8000 8000
- SELECT_COUNT_LUT: Selects one of the four packet-switched counters.

  For example:

```
CSL_aifHwControl (hAifLink, CSL_AIF_CMD_PE_MOD_LINK_84CNT_LUT, &a84CntLut);
```

The ID_LUT has the following fields to be programmed:

- AGGR_CTL_LUT: Controls aggregator mode. Should be insert for control slots. Add 7/8 or 15/16 bit in case of data slots.
- OBSAI_ADR_LUT: OBSAI address to be inserted in the message header.
- OBSAI_TYPE_LUT: OBSAI type to be inserted in the message header.

  For example:

```
CSL_aifHwControl (hAifLink, CSL_AIF_CMD_PE_MOD_LINK_21CNT_ID_LUT0, &aIdLut0);
```

- TX_RULE_MASK_LUT: Indicates which counter bits should be compared. A value of zero passes the compare.
- TX_RULE_MCOMPARE_LUT: Indicates the compare value. If the user intends not to have any empty data message slots, then the both the compare and the mask fields should be zero.

  For example:

```
CSL_aifHwControl (hAifLink, CSL_AIF_CMD_PE_MOD_LINK_21CNT_ID_LUT1, &aIdLut1);
```

- TERM_CNT_CFG: This register programs the terminal counts for 4 packet-switched counters. If you do not intend to have empty control message slots, the terminal counters should be set to zero.

7.1.5.2 Aggregator

The aggregator performs the following functions:

- Pass through of messages
  - Received messages via the combiner/decombiner
- Aggregate (sum payloads) downlink data at 16 or 15 bits per I and Q
- Aggregate (sum payloads) MAI data at 8 or 7 bits per I and Q for PIC
- Saturation logic for aggregation (I and Q saturated separately)
- OBSAI message insertion
  - Messages created by the protocol encoder, using transmission rules, may be legally inserted into the stream. A corresponding empty message must exist from the CD. Although the concept of message insertion does not exist for CPRI, the AG has the capability to insert or pass thru CPRI control words and AxC container data on a per-byte basis from either the CD or the PE.
- Error reporting (i.e., sum overflows; OBSAI header mismatches; and frame alignment)

The aggregator performs aggregation on a received link from the combiner and a created link from the protocol encoder. The aggregator does not aggregate streams from multiple DSP cores. The output of the aggregator feeds the Tx MAC. There are six aggregator blocks, one of each of the six link transmit/receive pairs.

There are several supported per-link options:

- A received link can be looped back out the transmit port (default). This is useful for a device pass thru mode or for a direct loopback for debug and test. Also, this mode would be used in a daisy chain to pass uplink data up the daisy chain.
- A created link may be sent out of the transmit port. It may be used for the start of a DL daisy chain and is used in conjunction with received empty messages from the combiner.
- The aggregator functionality is turned on for summation or message insertion.

The aggregation function is position-based, as opposed to address-based. Therefore, you must know the positional format of a received stream in order to properly align with the created stream from the protocol encoder (this is not required for simple pass-through of received data).

When used for message insertion, the aggregator can overwrite an empty message with a new message (this is useful in creating packet data messages). However, an empty message must exist from the combiner for the new message to be inserted. Otherwise, an error is issued and the message from the combiner is passed through. You can also insert received messages (from the combiner) using this mechanism. In this case, the protocol encoder must provide co-incident empty messages.

Based on transmission rules, signals provided by the protocol encoder can modify the aggregator functionality (e.g., AxC summation and control message insertion) on a per-message slot basis.

The aggregator supports either 7/15 bit IQ data or 8/16 bit IQ data. The aggregator sums the payloads of two 15/16 bit I and two 15/16 bit Q DL data messages together or two times two 7/8 bit I and two 7/8 bit Q (UL or MAI) messages together. If the result of summing is to overflow and sign wrap, saturation logic will limit the addition to the maximum positive or negative value and indicate that an overflow occurred for the specific AxC.

Table 26 shows an example of how two streams may be aggregated using control from the PE. Messages may be aggregated (summed), inserted, or allowed to pass through on a message-slot by message-slot basis.

Table 26. Aggregation Operation Example

From Protocol Encoder (PE)	PE Control to Aggregator	From Combiner / Decombiner (CD)	Aggregator Output
AxC1	A (aggregate)	AxC2	AxC1 + AxC2
Empty Message	I (insertion)	Empty Message	Empty Message
Empty Message	P (pass-thru)	AxC2	AxC2
AxC3	A	Empty Message	AxC3
Empty Message	P	Control Message	Control Message
Control Message	I	Empty Message	Control Message
Control Message1	I	Control Message2	Control Message2

Table 27. Aggregator Error Conditions

From Protocol Encoder	From Combiner / Decombiner	Data Result	Header Mismatch	Error Result
Any defined type	Empty message	Any defined type(1)	Not checked	None
Empty message	Any defined type	Any defined type(1)	Not checked	None
WCDMA/FDD	WCDMA/FDD	IQ sum	None	None
WCDMA/FDD	WCDMA/FDD	IQ sum	Address or timestamp	Error
WCDMA/FDD	Any type(2)	WCDMA/FDD (not summed)	Don't care	Error
Any type(2)	WCDMA/FDD	WCDMA/FDD (not summed)	Type	Error
Control message	Any type(3)	Any type(3)	Don't care	Error

(1) Any defined type such as insertion
(2) Cannot be empty message or WCDMA/FDD type
(3) Cannot be empty message type

Figure 46. Aggregator Configuration Input (1-of-6 links)

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		AGG_EN1	AGG_WIDTH_SEL	Reserved
R-0		R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 28. Aggregator Configuration Input (1-of-6 links) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	AGG_EN1		Enable aggregator
		0	Not enabled
		1h	Pass through from combiner
		2h	Pass through from protocol encoder
		3h	Aggregation function enabled
1	AGG_WIDTH_SEL		AxC sum width select (i.e., 15/16 bit vs. 7/8)
		0	7/15 bit widths
		1	8/16 bit widths
0	Reserved	0	Reserved

Example 3. Programming Aggregator Configuration Register

```
aAgCfg[0].aggrMode = CSL_AIF_AGGR_MODE_PE_IP_ONLY;
aAgCfg[0].bEnableAggrErr = TRUE;
#define AIF_AXC_OUT_AGGRCTRL_LINK_0 \
{     \

CSL_AIF_PE_AGGR_CTRL_ADD_15_16_BIT,  \
CSL_AIF_PE_AGGR_CTRL_ADD_15_16_BIT,  \
CSL_AIF_PE_AGGR_CTRL_ADD_15_16_BIT,  \
CSL_AIF_PE_AGGR_CTRL_ADD_15_16_BIT,  \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP      \
}

// Outbound AxC Aggregator Ctrl
   Uint8
aAxCAggrCtrlLut[AIF_NUM_ACTIVE_LINKS][CSL_AIF_MAX_NUM_CS_TRANS
MISSION_RULES] = {AIF_AXC_OUT_AGGRCTRL_LINK_0};

// aggr ctrl for ctrl data for link 0
#define AIF_CTRL_OUT_AGGRCTRL_LINK_0 \
{     \
CSL_AIF_PE_AGGR_CTRL_INSERT,  \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP,     \
CSL_AIF_PE_AGGR_CTRL_NOP      \
}
// Outbound Ctrl data Aggr. Ctrl
   Uint8
aCtrlAggrCtrlLut[AIF_NUM_ACTIVE_LINKS][CSL_AIF_MAX_NUM_CONTROL_T
RANSMISSION_RULES] =  {AIF_CTRL_OUT_AGGRCTRL_LINK_0};

for (i=0; i < CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES; i++ )
{
aPeCfg[0].aAggrCtrlAxCLut[i] = aAxCAggrCtrlLut[0][i];
aPeCfg[0].aAggrCtrlCtrlDataLut[i] = aCtrlAggrCtrlLut[0][i];
}
```

7.1.5.3 Tx MAC Configuration

You need to program the FIFO_THRESHOLD and the DELTA_OFFSET values. The FIFO_THRESHOLD is typically programmed to a value between 8 and 20. This value is programmed on the higher side for decombining, somewhere in the low and middle for combining, and on the lower side for everything else.

DeltaOffsetValue calculations are given in Appendix A. For information about transmission synchronization, see Section 7.3.4.

7.1.5.4 CO Configuration

The CPRI output data format converter configuration involves programming the data format for that particular link. The options are UL 7bit/8bit, DL 7bit/8bit. The CSL automatically configures these registers based on the information provided during the link configuration. Note that the configuration of the CO block is needed only in the case of CPRI mode.

7.1.6 Configure Receive Links

7.1.6.1 Address and Type Look-Up Tables

The OBSAI RP3 address field is 13 bits, but only 10 bits are used by the Protocol Decoder. For flexible support, any ten bits of the thirteen bits can be selected. These ten bits are programmable. The selected bits can be consecutive or have bit gaps. The mapped internal address bits must be indexed in order as shown with an example in Table 29. For example, mapping 3=>2, 2=>1, and 1=>0 is legal. Mapping 3=>0, 2=>1, and 1=>2 is not legal.

Table 29. Address Mapping Examples

External bits (12 to 0)	Selected? 0 = not selected 1 = selected	Internal bits (9 to 0)
12	0	-
11	1	9
10	1	8
9	1	7
8	1	6
7	1	5
6	1	4
5	1	3
4	0	-
3	1	2
2	1	1
1	1	0
0	0	-

The protocol decoder (PD) extracts messages from RX links and routes them to the appropriate inbound memory. Data flows into the protocol decoder as six RX links with sideband signals indicating K character. Extracted data flows to inbound memory structures. Inbound memory destinations include 3x packet-switched FIFOs, 6x circular circuit-switched data buffers, and one error FIFO for capture of messages which caused error conditions. In order to make this routing decision, PD inspects message header information using the type and address OBSAI RP3 fields. 32 programmable types and 1,024 programmable addresses are possible.

The type field defines whether the data is captured or discarded by a circuit switch RAM or a packet switch RAM. The settings to capture or discard each type apply to all links. You can also program the type field to cause an error/alarm condition and trigger the data to be captured by the error FIFO.

The address field defines whether the address is illegal and the illegal address can trigger an error/alarm condition. The address field defines which of the three packet-switch FIFOs will capture the packet-switch data. The three packet-switched FIFOs are intended to be used to target a specific GEM core. The address field defines which circuit-switch and/or packet-switch data that the error FIFO will capture. The address field is used for address generation for circuit-switch data. You can also use the address field to define the internal data format of the data.

The 10-bit address indexes into the protocol decoder address look-up table (See Section 8.2.11.7). This look-up table is programmed with antenna carrier index and internal data format. The antenna carrier index returned by this look-up table forms 4 of the bits in the circuit-switched RAM address.

Example 4. Populating TYPE and ADDRESS Field LUT Parameters

```
#define AIF_NUM_ACTIVE_AXS_INBND_LINK_0   4
#define AIF_NUM_TYPE_FIELD_ENTRIES        3
#define AIF_TYPE_FIELD_LUT \
{CSL_AIF_PD_OBSAI_TYPE_VALUE_PKT_SW, CSL_AIF_PD_OBSAI_TYPE_VALUE_INACTIVE,
 CSL_AIF_PD_OBSAI_TYPE_VALUE_CIR_SW}

#define AIF_AXC_ADDR_LUT_LINK_0 \
{     \
0x0,  \
0x1,  \
0x2,  \
0x3}

Uint8 aTypeFieldLut[AIF_NUM_TYPE_FIELD_ENTRIES] = AIF_TYPE_FIELD_LUT;

Uint16 aAxCAddrLut[AIF_NUM_TYPE_FIELD_ENTRIES] = AIF_AXC_ADDR_LUT_LINK_0;
```

Packet-switched data is identified by the OBSAI RP3 type field and it is captured to a series of input (Inbound) FIFOs. The PD maintains an address look-up table for the type field (this will indicate if the message should be stored in either the circuit-switched buffer or the packet-switch buffer). The PD also maintains an address look-up table for the address field (this will specify which of three inbound FIFOs to store the packet-switched message). All 19 bytes of the OBSAI RP3 message are captured to the FIFOs.

Example 5. Populating PS Inbound FIFOs Parameters

```
#define AIF_PS_ADDR_LUT_FIFO_0 \
{     \
0x40  \
}
   // Inbound PS FIFO Addr LUT
   Uint16
aPsAddrLut[AIF_NUM_ACTIVE_INBND_PS_FIFOS][AIF_MAX_SIZE_INBND_PS_
ADDR_FIFO] =
{AIF_PS_ADDR_LUT_FIFO_0};
```

Example 6. Populating Parameters for PD

```
// PD common cfg
   pdCommonCfg.addressMask = 0x3FF;
   pdCommonCfg.numTypeFieldEntries = AIF_NUM_TYPE_FIELD_ENTRIES;
   pdCommonCfg.pInboundTypeFieldLut = aTypeFieldLut;
   pdCommonCfg.sizeInboundPsAddrFieldLut[0] =
AIF_SIZE_INBND_PS_ADDR_FIFO_0;
   pdCommonCfg.sizeInboundPsAddrFieldLut[1] =
AIF_SIZE_INBND_PS_ADDR_FIFO_1;
   pdCommonCfg.sizeInboundPsAddrFieldLut[2] =
AIF_SIZE_INBND_PS_ADDR_FIFO_2;
   pdCommonCfg.sizeInboundPsAddrFieldLut[3] =
AIF_SIZE_INBND_PS_ADDR_FIFO_3;
   pdCommonCfg.pInboundPsAddrFieldLut[0] = &aPsAddrLut[0][0];
   pdCommonCfg.pInboundPsAddrFieldLut[1] = NULL;
   pdCommonCfg.pInboundPsAddrFieldLut[2] = NULL;
   pdCommonCfg.pInboundPsAddrFieldLut[3] = NULL;
// populate PD link fields
   aPdCfg[0].bEnablePd = TRUE;
   aPdCfg[0].bCpriCtrlWordCapture = FALSE;
   aPdCfg[0].pPdCommonSetup =
   aPdCfg[0].sizeInboundAxCAddrFieldLut =
```

Example 6. Populating Parameters for PD (continued)

```
AIF_NUM_ACTIVE_AXS_INBND_LINK_0;
   aPdCfg[0].pInboundAxCAddrFieldLut = &(aAxCAddrLut[0][0]);
```

For CPRI mode, a dedicated Stream Index LUT is used instead of the address LUT. The raw CPRI AxC is used as an index into this LUT and the LUT returns stream index [0-16] and the data format. For Control words in CPRI, user can choose to either capture all the control words or discard all of them for a particular link by setting or resetting the CPRI Packet switch enable bit for the particular link.

Example 7. CPRI Mode Code

```
#define AIF_CPRI_SI_AXC_LUT \
{ 0x0, 0x1, 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, 0xA, 0xB, 0xC, 0xD, 0xE, 0xF }

Uint16 aAxCCpriSiLut [CSL_AIF_CPRI_MAX_NUM_AXC_PER_4X_LINK_7_15_BIT] =
    AIF_CPRI_SI_AXC_LUT;
// populate PD link fields
aPdCfg[0].bEnablePd = TRUE;
aPdCfg[0].bCpriCtrlWordCapture = TRUE;
aPdCfg[0].pInboundCpriSiAxCLut = aAxCCpriSiLut;
```

7.1.6.2 PE and PD Circuit-Switched Address Generation

Three important factors shape the way circuit switched address is generated:

- Internal Format to be supported {DL, RSA UL, PIC MAI, Generic}
- Endianess {Big Endian, Little Endian}
- VBUS Burst length {4 Chip Burst, 8 Chip Burst}

The circuit-switched RAM address bits map as follows:

Adr[13:12]	: Link#
Adr[10:4]	Quadword address
Adr[10:9]	8 Burst address
Adr[10:8]	4 Burst address
Adr[3:0]	16 bytes within a quadword

Table 30 and Table 31 show circuit-switched address generation for burst of 4 and burst of 8, respectively.

Table 30. PE and PD Circuit-Switched Address Generation (4 Burst)

	OBSAI				CPRI			
Address	DL	UL RSA	Generic	PIC MAI	DL	UL RSA	Generic	PIC MAI
Adr[13]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]
Adr[12]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]
Adr[11]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]
Adr[10]	TS[2]	TS[2]	TS[2]	TS[2]	TS[4]	TS[4]	TS[4]	TS[4]
Adr[9]	TS[1]	TS[1]	TS[1]	TS[1]	TS[3]	TS[3]	TS[3]	TS[3]
Adr[8]	TS[0]	BI[0]	TS[0]	TS[0]	TS[2]	BI[0]	TS[2]	TS[2]
Adr[7]	SI[3]	SI[3]	SI[3]	SI[3]	SI[3]	SI[3]	SI[3]	SI[3]
Adr[6]	SI[2]	SI[2]	SI[2]	SI[2]	SI[2]	SI[2]	SI[2]	SI[2]
Adr[5]	SI[1]	SI[1]	SI[1]	SI[1]	SI[1]	SI[1]	SI[1]	SI[1]
Adr[4]	SI[0]	SI[0]	SI[0]	SI[0]	SI[0]	SI[0]	SI[0]	SI[0]
Adr[3]	BI[2]	TS[0]	BI[2]	BI[2]	TS[1]	TS[2]	TS[1]	TS[1]
Adr[2]	BI[1]	BI[2]	BI[1]	BI[1]	TS[0]	TS[1]	TS[0]	TS[0]
Adr[1]	BI[0]	BI[1]	!BI[0]	BI[0]	BI[0]	TS[0]	!BI[0]	BI[0]
Adr[0]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 31. PE and PD Circuit-Switched Address Generation (8 Burst)

	OBSAI				CPRI			
Address	DL	UL RSA	Generic	PIC MAI	DL	UL RSA	Generic	PIC MAI
Adr[13]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]	LI[2]
Adr[12]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]	LI[1]
Adr[11]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]	LI[0]
Adr[10]	TS[2]	TS[2]	TS[2]	TS[4]	TS[4]	TS[4]	TS[4]	TS[4]
Adr[9]	TS[1]	TS[1]	TS[1]	TS[3]	TS[3]	TS[3]	TS[3]	TS[3]
Adr[8]	SI[3]	SI[3]	SI[3]	TS[2]	SI[3]	SI[3]	SI[3]	SI[3]
Adr[7]	SI[2]	SI[2]	SI[2]	SI[3]	SI[2]	SI[2]	SI[2]	SI[2]
Adr[6]	SI[1]	SI[1]	SI[1]	SI[2]	SI[1]	SI[1]	SI[1]	SI[1]
Adr[5]	SI[0]	SI[0]	SI[0]	SI[1]	SI[0]	SI[0]	SI[0]	SI[0]
Adr[4]	TS[0]	BI[0]	TS[0]	SI[0]	TS[2]	BI[0]	TS[2]	TS[2]
Adr[3]	BI[2]	TS[0]	BI[2]	BI[2]	TS[1]	TS[2]	TS[1]	TS[1]
Adr[2]	BI[1]	BI[2]	BI[1]	BI[1]	TS[0]	TS[1]	TS[0]	TS[0]
Adr[1]	BI[0]	BI[1]	!BI[0]	BI[0]	BI[0]	TS[0]	!BI[0]	BI[0]
Adr[0]	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

These abbreviations are used in Table 30 and Table 31:

BI	Dual Byte Index	PE always treats payload bytes in pairs. These dual payload bytes are enumerated by the PE 0-31 (where 0 represents the first double byte received in a OBSAI/CPRI payload) and used extensively for memory addressing. The OBSAI/CPRI FSM supplies this Dual Byte Index.
LI	Link Index	There are six SERDES links, and a corresponding PE Link per SERDES link. The links are enumerated 0-5.
SI	Stream Index	Up to 16 streams are supported per link. For PE, the 84 LUT gives the stream index while for PD, the SI is supplied by either the (OBSAI mode) address LUT or (CPRI) the CPRI SI LUT.
TS	Time Stamp	The Time Stamp counter increments once every four OBSAI chips or once every one CPRI chip. Time stamp is a 6-bit counter. Some of these bits are used for address generation.

7.1.6.3 CI Configuration

The CPRI input data format converter configuration involves programming the data format for that particular link. The options are UL 7bit/8bit, DL 7bit/8bit. The CSL automatically configures these registers based on the information provided during the link configuration. Note that the configuration of the CI block is needed only in the case of CPRI mode.

7.1.6.4 Rx MAC Configuration

The RX Mac configuration is explained in Section 7.3 with Rx link synchronization.

7.1.6.5 DB Configuration

The address mappings of different Data Buffer RAMs and FIFOs are detailed in Table 32.

Table 32. Data Buffer Address Mapping

Address	Access	Description
0xA0xx_xxxx	R/W	Inbound circuit-switched RAM (links 0-3)
0xA1xx_xxxx	R/W	Inbound circuit-switched RAM (links 4-5)
0xA2xx_xxxx	R/W	Outbound circuit-switched RAM (links 0-3)
0xA3xx_xxxx	R/W(1)	Outbound circuit-switched RAM (links 4-5)
0xA4xx_xxxx	R	Inbound packet-switched FIFO (OBSAI access)
0xA5xx_xxxx	W	Outbound packet-switched FIFO (OBSAI access)
0xA6xx_xxxx	R/W	Inbound packet-switched RAM (CPRI/debug access)
0xA7xx_xxxx	R/W	Outbound packet-switched RAM (CPRI/debug access)
0xA8xx_xxxx	W	DMA completion registers
0xA9xx_xxxx	R/W	Data trace capture RAM
0xAAxx to 0xXf	--	Reserved

(1) Outbound circuit-switched RAM is unique. It is primarily written, but also read for use with delayed streams.

These terms are used for all data buffer address mapping:

Circuit-switched data— Typically antenna data with regular flow characteristics. Header information address and type are expected to be static for each antenna carrier.

Packet-switched data— Typically control information with irregular traffic flow characteristics. Header information is highly variable and handled by SW.

Inbound— Data or control information that has or will **enter** the TMS320TCI6487/8

Outbound— Data or control information that has or will **exit** the TMS320TCI6487/8

RAM block— A 32-chip portion of RAM dedicated for a particular SERDES link.

Block_size— 2k bytes = 16 AxC x 32 chips x 4 byte/chip (4 or 8 Segments)

RAM segment— The RAM block is further divided into 4- or 8-chip portions called segments. Each data buffer RAM location holds 4 chips (quadword), so an 8-chip RAM segment has two consecutive memory locations per stream.

Segment_size— (UL) 512 bytes = 16 AxC x 8 chips x 4-byte/chip
(DL) 256 bytes = 16 AxC x 4 chips x 4-byte/chip

Table 33. VBUS Circuit-Switched RAM Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (0xA in case of TMS320TCI6488).
Adr[27:25]		Specifies circuit-switched RAM
	3'b000	Inbound circuit-switched RAM
	3'b001	Outbound circuit-switched RAM
Adr[24:22]		Link number:
	3'b000	Link 0
	3'b001	Link 1
	3'b010	Link 2
	3'b011	Link 3
	3'b100	Link 4
	3'b101	Link 5
	3'b110	Unpopulated address(1)
	3'b111	Unpopulated address
Adr[21:11]		Unpopulated. Allows 64K chip wrap.(2)
Adr[10:4]		Quadword address within a link. See Section 7.1.6.2.
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

(1) The circuit-switched RAM is built from three banks representing the first three address portions.
(2) AIF addresses must wrap in order to work within TPDMA limitations. This allows TPDMA to increment beyond the 32-chip buffer. Internally, the buffer wraps thus reducing the number of DMA param entries.

Table 34. VBUS Inbound Packet-Switched FIFO Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies packet-switched FIFO
	4'b0100	Inbound packet-switched FIFO
	4'b0101	Outbound packet-switched FIFO
Adr[23:22]		FIFO number:
	2'b00	Inbound FIFO #0
	2'b01	Inbound FIFO #1
	2'b10	Inbound FIFO #2
	2'b11	Inbound error FIFO
Adr[21:5]		Unpopulated/unused. Allows the FIFO to be VBUS accessible without needing to use the VBUS FIFO mode.
Adr[4]		Quadword pair address
	1'b0	Low quadword, OBSAI message payload
	1'b1	High quadword, 3 byte OBSAI header or read causes FIFO address to increment.
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

Table 35. VBUS Outbound Packet-Switched FIFO Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies packet-switched FIFO
	4'b0100	Inbound packet-switched FIFO
	4'b0101	Outbound packet-switched FIFO

Table 35. VBUS Outbound Packet-Switched FIFO Addressing (continued)

Address	Value	Description
Adr[23:19]		FIFO number:
	5'h00	Outbound FIFO #0
	5'h01-5'h28	Outbound FIFO #1-28
	5'h29	Outbound FIFO #29
	5'h1e-5'h1f	Reserved/unpopulated
Adr[18:5]		Unpopulated/unused. Allows the FIFO to be VBUS accessible without needing to use the VBUS FIFO mode.
Adr[4]		Quadword pair address
	1'b0	Low quadword, OBSAI message payload or 16 bytes of CPRI control words
	1'b1	High quadword, 3 byte OBSAI header or read causes FIFO address to increment.
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

In Table 36, packet-switched RAM is the same as RAM used for packet-switched FIFOs. Either FIFO or RAM access is supported in data buffering.

Table 36. VBUS Outbound Packet-Switched RAM (OBSAI PktSw Format) Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies packet-switched RAM
	4'b0110	Inbound packet-switched RAM
	4'b0111	Outbound packet-switched RAM
Adr[23:14]		Unpopulated/unused. Causes RAM to wrap addresses.
Adr[13:5]		512 pairs of quadword RAM addresses
Adr[10:4]		Quadword pair address
	1'b0	Low quadword, OBSAI message payload or 16 bytes of CPRI control words
	1'b1	High quadword, 3 byte OBSAI header or unused for CPRI.
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

Table 37 represents the same RAM and VBUS address range as is used in OBSAI mode. However, the address bits are used in a different way.

Table 37. VBUS Packet-Switched RAM (CPRI Control Word Format) Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies packet-switched RAM
	4'b0110	Inbound packet-switched RAM
	4'b0111	Outbound packet-switched RAM
Adr[23:14]		Link number:
	3'b000	Link 0
	3'b001	Link 1
	3'b010	Link 2
	3'b011	Link 3
	3'b100	Link 4
	3'b101	Link 5
	3'b110-3'b111	Unused
Adr[13:11]		Unpopulated

Table 37. VBUS Packet-Switched RAM (CPRI Control Word Format) Addressing (continued)

Address	Value	Description
Adr[10:5]		Quadword RAM addresses. CPRI Control Words are stored sequentially starting with lowest address to highest address.
Adr[4]		Header
	0'b0	Always set to zero in CPRI mode; there is no header.
	1'b0	In OBSAI mode, represents Header versus Payload.
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

DMA registers are write only. The DMA writes to the Table 38 locations at the end of DMA bursts. These registers reside on the DMA/Data VBUS, not on the configuration bus.

Table 38. VBUS DMA Completion Registers Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies DMA completion registers
	4'b0100	DMA completion registers
	4'b0101	Unused
Adr[23:20]		FIFO number
	4'h0	Inbound Link 0, DMA Complete
	4'h1	Inbound Link 1, DMA Complete
	4'h2	Inbound Link 2, DMA Complete
	4'h3	Inbound Link 3, DMA Complete
	4'h4	Inbound Link 4, DMA Complete
	4'h5	Inbound Link 5, DMA Complete
	4'h6-4'h7	Reserved/unpopulated
	4'h8	Outbound Link 0, DMA Complete
	4'h9	Outbound Link 1, DMA Complete
	4'ha	Outbound Link 2, DMA Complete
	4'hb	Outbound Link 3, DMA Complete
	4'hc	Outbound Link 4, DMA Complete
	4'hd	Outbound Link 5, DMA Complete
	4'he-4'hf	Reserved/unpopulated
Adr[21:11]		Unpopulated
Adr[10:4]		Quadword address
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

Table 39. Data Trace Capture RAM Addressing

Address	Value	Description
Adr[31:28]		Specifies AIF module (decode at higher level).
Adr[27:24]		Specifies data trace capture RAM
	4'b1000	Unused
	4'b1001	Data trace capture RAM
Adr[23:10]		Unpopulated/unused. Allows VBUS RAM address to increment beyond the populated address range while local RAM wraps.
Adr[9:4]		Ram quadword address
Adr[3:0]		Byte address 0-15. Bytes are not VBUS addressable.

7.1.6.5.1 Configuration for Packet-Switched Data

Packet-switched data is a messaging mechanism used to pass control data between multiple TMS320TCI6487/8 and different boards over the antenna interface. The Protocol Decoder extracts this data from incoming OBSAI links based on header address and header type information. Four FIFOs are supported in the inbound direction whereas 30 FIFOs are provided in the outbound direction. On the outbound side, the choice of a particular FIFO is done by programming the PE_LINK_ID_LUT0 (see Section 8.2.12.8) for each of the six links. On the inbound side, capture to any of these FIFO is controlled by LUT bits. Capture is enabled if Mode = OBSAI and any of the following conditions are met:

- Type=Packet-switched data
- Type=Error
- Type=Circuit-switched data and AXC capture enable is activated.
- Time stamp failed time stamp check

If the first condition is met (Type= PktSw), then the message is captured to the FIFO number corresponding to the PKT_FIFO_INDEX_LUT fields found in the PD_ADR_LUT register (see Section 8.2.11.7). For the other conditions, the message is captured to FIFO #3, the Error FIFO.

Note: Normal OBSAI 1x links provide for a control slot once out of every 21 messages. Because of Combining, four of these 1x OBSAI links could be represented by a single AIF 4x link. It is important to direct control data to the appropriate control slot; a FIFO is dedicated for each possible control slot. An additional FIFO is dedicated for all possible message slots for a given link bringing the total number of FIFO structures up to 5x per outbound link or 30x total outbound FIFOs.

RAM is used in both the inbound and outbound directions to create these FIFO like structures. Two RAMs 512 x 19Byte are used for this purpose. The user has flexibility in how these RAMs are used for this purpose. In the inbound direction, the user has to program how much RAM is allocated for the four different FIFOs. In the outbound direction, a single RAM location is allocated for an outbound message regardless of which of the 30 FIFOs it is destined for. In CPRI mode, the FIFO circuitry is un-used and the RAM addressed directly via the DMA/data VBUS. FIFO Size: The inbound FIFO sizes are configured by programming the DB_IN_SIZE_CFG register. The RAM is partitioned via writing 3 programmable RAM partition fields. Each of these values are 5 bits and correspond to the 5 Msbs of the inbound packet-switched, physical RAM address. This configuration granularity allows allocating n x 16 RAM locations the different FIFO.

The supported FIFOs are {FIFO#0, FIFO#1, FIFO#2, ErrFIFO}. The beginning and ending memory addresses for the FIFOs are as follows:

- FIFO#0 Start = 0x000
- FIFO#1 Start = InFifoRamPart0 x 16
- FIFO#2 Start = InFifoRamPart1 x 16
- ErrFIFO Start = InFifoRamPart2 x 16
- FIFO#0 End = (InFifoRamPart0 x 16) - 1
- FIFO#1 End = (InFifoRamPart1 x 16) - 1
- FIFO#2 End = (InFifoRamPart2 x 16) - 1
- ErrFIFO End = 0x1ff

For example:

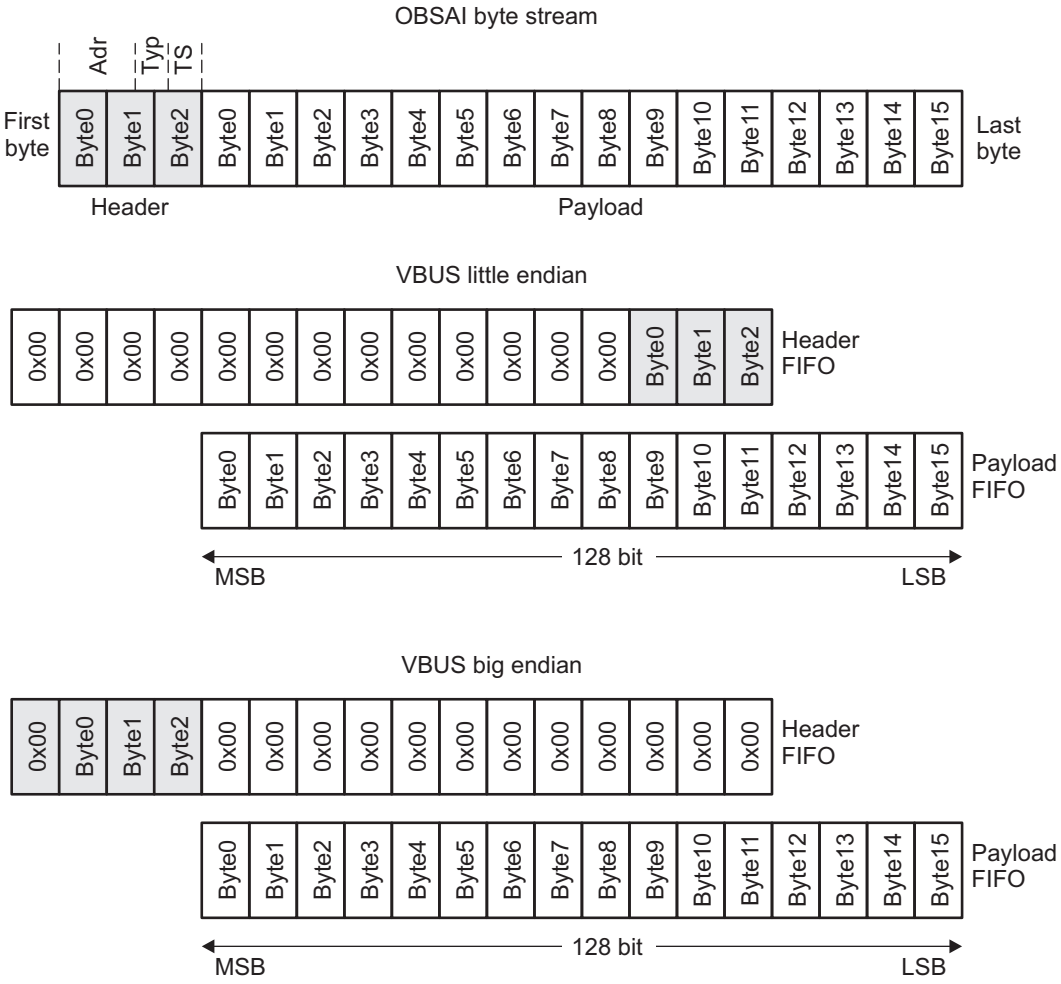
```
CSL_AifInboundFifoSetup psInFifoCfg;
gblCfg.pInboundPsFifoSetup = &psInFifoCfg;
psInFifoCfg.sizeFifo[0]  = 1
psInFifoCfg.sizeFifo[1]  = 1
psInFifoCfg.sizeFifo[2]  = 1
psInFifoCfg.sizeFifo[3]  = 1
...
```

Note: It is possible to allocate no RAM for a given FIFO. In this case, the FIFO is not usable and will flag an overflow/underflow alarm condition anytime access is attempted. For legal configuration of the FIFO,

- InFifoRAMpart1 must be greater or equal to InFifoRamPart0
- InFifoRAMpart2 must be greater or equal to InFifoRamPart1

Figure 47 illustrates the structure of the inbound packet-switched memory.

Figure 47. Inbound Packet-Switched Memory Structure



The DSP or TPDMA sends outbound packet-switched messages to the AIF over the 16-byte VBUS. Messages arrive in a minimum of two VBUS cycles. Each message is written to the 19 byte-wide outbound circular RAM. The data/DMA VBUS is 16 bytes wide while the packet-switched RAM is 19 bytes wide. A minimum of two vbus writes (or reads) is required for every RAM/FIFO access. The first write is the 16-byte payload which occupies the 16 LSBytes of the packet-switched RAM. The second write is the 3-byte header which occupies the 3 MSBytes of the packet-switched RAM. When reading (or writing) the OBSAI FIFOs, the payload portion is always read first. The payload format is identical for both big and little endian. The header portion is word swapped for big and little endian. The circuit maintains two pointers for each of the 30x possible FIFOs. These pointers are:

Head pointer— Points to the next read location for the given FIFO. This is the read location for both the Linked List RAM and the Packet RAM.

Tail pointer— Points to the last written location of the Packet RAM. Also, the same address points to the next location in the Linked List RAM which is updated on the next write.

Associated with a head and tail pointer pair is a not_empty bit which indicates whether the FIFO is empty or not empty. At startup, this bit is cleared indicating the FIFO is empty and also indicating that the head and tail pointers may currently contain random data. After any write to a given FIFO the bit changes to indicate not_empty. There are two conditions where the bit changes to the empty state:

- Last read: head & tail pointers are equal, and a read is performed.
- Reset/flush: MMR Write, HW reset, or error condition which flushes FIFO or MMR bit indicating FIFO flush.

The DB circuitry continuously checks for overflow conditions. Overflow occurs when any of the write pointers is equal to any of the head pointers, the FIFO not_empty bit is set and a VBUS write is attempted. Whenever a VBUS write occurs, all head pointers are re-evaluated against the current write pointer to check for the overflow condition.

The DSP core may service the arrival of packet-switched data using interrupts to the SW. In this case, the every arrival of packet-switched data will strobe the system event to the DSP core. Another scenario is that a TPDMA channel is set up to transfer FIFO contents with some Burst interval such as a burst of 16 messages.

There are seven system events associated with the inbound packet-switched FIFOs (four not_empty flags and three programmable_not_empty flags). Each of the FIFOs have a dedicated system event (not_empty) which is active high whenever the FIFO is not empty. Additionally, FIFOs {0, 1, 2} each have a system event (programmable_not_empty) which strobes (active high for 1 vbus clock cycle) when n writes to the FIFO have occurred. The value n can be programmed in the DB_IN_FIFO_EVNT_CFG register.

For example:

```
CSL_AifInboundFifoSetup psInFifoCfg;
psInFifoCfg.eventDepth[0] = CSL_AIF_INBOUND_PS_FIFO_EVENT_DEPTH_2;
psInFifoCfg.eventDepth[1] = CSL_AIF_INBOUND_PS_FIFO_EVENT_DEPTH_2;
psInFifoCfg.eventDepth[2] = CSL_AIF_INBOUND_PS_FIFO_EVENT_DEPTH_2;
psInFifoCfg.eventDepth[3] = CSL_AIF_INBOUND_PS_FIFO_EVENT_DEPTH_2;
gblCfg.pInboundPsFifoSetup = &psInFifoCfg;
...
```

For start-up initialization, debug, and CPRI uses, the FIFO's RAM is directly accessible through VBUS direct random access. This offline, direct access is achieved through alternate addresses (see table). This provides the DSP a test/debug port into the RAM and also allows the DSP to use the RAM in applications where the AIF is unused.

Example 8. Trigger for EDMA to Transfer Packet-Switched Data from DSP to AIF Outbound Packet-Switched FIFO

```
# define SAMPLE_COUNT_WRAP_WCDMA_FDD 7
//Assuming the FSYNCCLK/ALTFSYNCCLK is 30.72 MHz
/** configure mask based trigger event to occur every 20 chips for EDMA x'fer
(ExtMem -> AIF PS FIFO 0)* events occur 5,25,45..*/

configCounterTrigger[0].timerUsed = CSL_FSYNC_RP3_TIMER;
configCounterTrigger[0].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_10;
configCounterTrigger[0].eventCount=20*(SAMPLE_COUNT_WRAP_WCDMA_FDD+1)-1;

configCounterTrigger[0].offset.slotOffset = 0;
configCounterTrigger[0].offset.chipTerminalCountIndex = 0;
configCounterTrigger[0].offset.chipOffset = 5;
configCounterTrigger[0].offset.sampleOffset = 0;
```

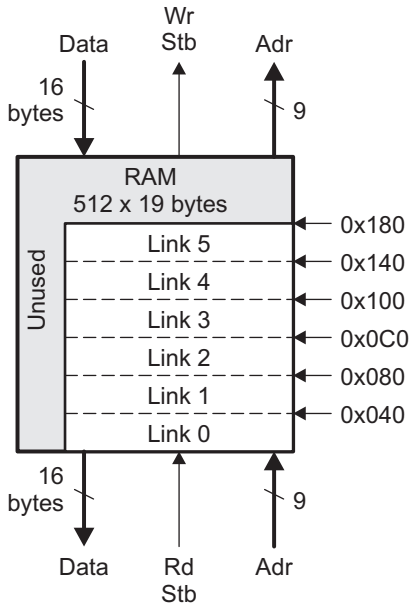
Note: For the transfer of packet-switched data from AIF to DSP/External memory, you can either configure a frame sync event to trigger the transfer or use the AIF generated events 40, 41 and 42 to trigger the transfer.

7.1.6.5.2 Data Buffer CPRI Mode, Packet-Switched Memory

CPRI control words have very regular/deterministic flow characteristics very similar AxC data. For this reason, the CPRI control words are handled with circular indexed RAM. Unlike OBSAI, there is not CPRI addressing scheme rather all CPRI control words are extracted and sent to the GEM cores. The user should set up the EDMA transfers to transfer CPRI control data from/to GEM/External memory.

In CPRI mode, the RAM has a fixed partition on a link by link basis and only 16 bytes of the RAM's width is utilized. The fix partition allocates 64 physical RAM addresses for each of 6 links. With a width of 16 bytes per RAM address, a total of 1024 bytes are allocated per link. The 3 MSB of the 19 Byte wide RAM and RAM addresses 0x180-to-0x1ff are un-utilized in CPRI mode.

Figure 48. Packet-Switched RAM Allocation for CPRI Mode



As CPRI control words arrive in the Protocol Decoder, they are captured sequentially and packed into 16 byte quadwords. These quadwords are then written into the appropriate RAM partition for the given link. The PD circular indexes the RAM partition, incrementing the address after every quadword write. There are a total of 256 CPRI control words in a CPRI hyperframe. Each of these control words is {1, 2, 4} bytes corresponding to {1x, 2x, 4x} link rates. As a result of this difference, the allocated RAM per segment holds {4, 2, 1} hyperframes of CPRI control words corresponding to {1x, 2x, 4x} link rates.

Note: Control Byte0 is always the MSB of the CPRI control word RAM, irrespective of whether the device is in big-Endian or little-Endian mode.

Note: It is possible that your application does not use the CPRI in the order in which the AIF has captured the control data. The application must reorder the CPRI conrol words so that CPRI sub-channels can be extracted.

7.1.7 Configure Combiner/Decombiner

The combiner / decombiner is essentially an intelligent multiplexer / demultiplexer. Links are combined or decombined using a round robin algorithm. Also, the RX (source) of one link may be redirected to the TX (destination) of another link. Combining is typically used in the uplink direction on multiple slower links originating from the backplane. Decombining is typically used in the downlink direction to drive multiple slower links on the backplane. There are two combiner and two decombiner functions in CD. Each is programmed separately. There are six C/D multiplexer outputs corresponding to each link.

7.1.7.1 Redirection

Redirection is also supported in CPRI mode. Any input link can be directed to any output link of the same rate. One input link can go to only one output link. Based on link configuration information, the CD knows the rate of each link and whether that link is enabled. What it needs to know is the source link and destination link. This is programmed by way of the configuration register CD_OUT_MUX_SEL_CFG which has two fields (see Section 8.2.7.1). One is the source select field and the other is input enable field. For the source select field, values 0 through 5 represent the link number to be routed. The input enable field should be enabled for the link that is used as the source. For example, if link1 needs to be redirected to link2, then the source select field for link2 should be set to 1 and the input enable field for link1 should be set.

For example:

```
cdSetup.bCdEnableMask[1] = TRUE;
CdSetup.cdOutSrcSel[2] = CSL_AIF_LINK_1;
pGblCfg.pCdSetup = &cdSetup
```

The input enable can be disabled for the links that are not used as inputs for redirection/combining and decombining.

7.1.7.2 Combining

There are two separate combining units. Each combiner will create the resulting link frame structure, including Idle bytes. The Combiner will also insert empty messages when:

- Fewer links are to be combined than the resulting link can support (e.g. 1 - 1x link into a 4x link).
- A link feeding the Combiner goes down.

Combining (multiplexing) is done using the round-robin algorithm. In combining, all of the incoming messages are treated the same and just placed one after another. If four 1x links are combined, the operation goes as this: First message from input link 1, then a message from link 2, a message from link 3, a message from link 4 and again from 1, 2, 3...

The links that are combined must be frame aligned. Each of the two combiners provides its own programmable offset and valid data window. The master frame boundary (K28.7 byte) should fall within the valid data window following the combiner offset-relative to the frame sync. For delta calculations of the combined output link, see Section A.1. You have to program a four location look-up table (see Section 8.2.7.2) which determines the source links to be combined as well as the order that they are combined. The table must always be complete as the combiner will cycle through the entire table. If there are fewer links to be combined than will fill the output link, empty links must be used as table entries. You also need to program the source select field (in CD_OUT_MUX_SEL_CFG register) for the combined output link with a value of 6 or 7 depending on whether combiner 0 or combiner 1 is being used. See Table 40 for examples on how links may be combined.

Table 40. Link Combining Examples

Links to be Combined		Resulting Output Link	
Rate	Link Number	Rate	Combining Sequence
1x 1x	0 3	2x	0, 3, 0, 3, ...
1x 1x 1x	0 2 4	4x	0, 2, 4, E, 0, 2, 4, E, …
1x 1x 2x	2 4 5	4x	2, 5, 4, 5, 2, 5, 4, 2, …
2x 1x	1 3	4x	3, 1, E, 1, 3, 1, E, 1, …
2x 2x	0 5	4x	0, 5, 0, 5, ...
2x	0	4x	0, E, 0, E, ...
2x	0	4x	E, 0, E, 0 …
2x	0	4x	0, 0, E, E …
2x	0	4x	E, E, 0, 0 …
1x	0	4x	0, E, E, E, 0, E, E, E, 0, …

Example 9. Combine Link 0 and Link 1 Into Link 2

```
//Enable masks of links
cdSetup.bCdEnableMask[0] = TRUE;
cdSetup.bCdEnableMask[1] = TRUE;
cdSetup.bCdEnableMask[2] = TRUE;
cdSetup.bCdEnableMask[3] = FALSE;
cdSetup.bCdEnableMask[4] = FALSE;
cdSetup.bCdEnableMask[5] = FALSE;
cdSetup.pCombinerSetup[0] = &aCbSetup[0];
//Select combiner zero

cdSetup.cdOutSrcSel[2] = CSL_AIF_CD_OUT_SRC_CB_0;
aCbSetup[0].bCombinerEnable  = TRUE;
aCbSetup[0].combinerInput[0] = cSL_AIF_LINK_0;
aCbSetup[0].combinerInput[1] = CSL_AIF_LINK_1;
aCbSetup[0].combinerInput[2] = CSL_AIF_LINK_0;
aCbSetup[0].combinerInput[3] = CSL_AIF_LINK_1;
aCbSetup[0].frameSyncOffset  = 600;
aCbSetup[0].windowMasterFrameOffset = 19;
gblCfg.cdSetup = &cdSetup
```

7.1.7.3 Decombining

There are two separate decombining units. The Decombiner is essentially the reverse function of the combiner. The Decombiner creates the resulting link frame structures, including idle bytes. The Decombiner can drop messages on the floor if fewer destination links are requested than the source link supports. Input links are decombined in a round robin order; empty links are required as place holders by the protocol encoder for unused message slots.

Table 41. Link Decombining Examples

Link to be Decombined		Resulting Output Links		
Rate	**Input Sequence**	**Rate**	**Link Number**	**Output Sequence**
2x	a, b, c, d, e, f, ...	1x 1x	0 3	a, c, e, ... b, d, f, ...
4x	a, b, c, **d**, e, f, g, **h**, i, j, k, ...	1x 1x 1x	0 2 4	a, e, i, ... b, f, j, ... c, g, k, ...
4x	a, b, c, d, e, f, g, h, i, j, k, ...	1x 1x 2x	2 4 5	a, e, i, ... c, g, k, ... b, d, f, h, j, ...
4x	a, b, **c**, d, e, f, g, h, i, j, **k**, ...	2x 1x	1 3	b, d, f, h, j, ... a, e, i, ...
4x	a, b, c, d, e, f, g, h, i, j, k, ...	2x 2x	0 5	a, c, e, ... b, d, f, ...
4x	a, **b**, **c**, **d**, e, **f**, **g**, **h**, i, **j**, **k**, ...	1x	0	a, e, i, ...

The means to select destination links for decombining is a 4-location look-up table (see Section 8.2.7.6). The look-up table determines the destination links to decombine into the selected source link, as well as the decombining order. Using a simple counter as a pointer, the table is accessed in round-robin fashion. The table must always be complete as the Decombiner cycles through the entire table. If there are fewer destination links for decombining than originated from the source link, empty links must be used as table entries.

The decombiner always starts with the first entry in the table. When fewer output links are selected than the input link supports, empty link entries in the destination table tell the Decombiner to that message. This is illustrated in Table 41 using bold letters in the Input Sequence column to show the skipped source messages. The source select field (in CD_OUT_MUX_SEL_CFG register) for the decombined output links should be programmed with the same link as in CD_DC_SRC_SEL_CFG register.

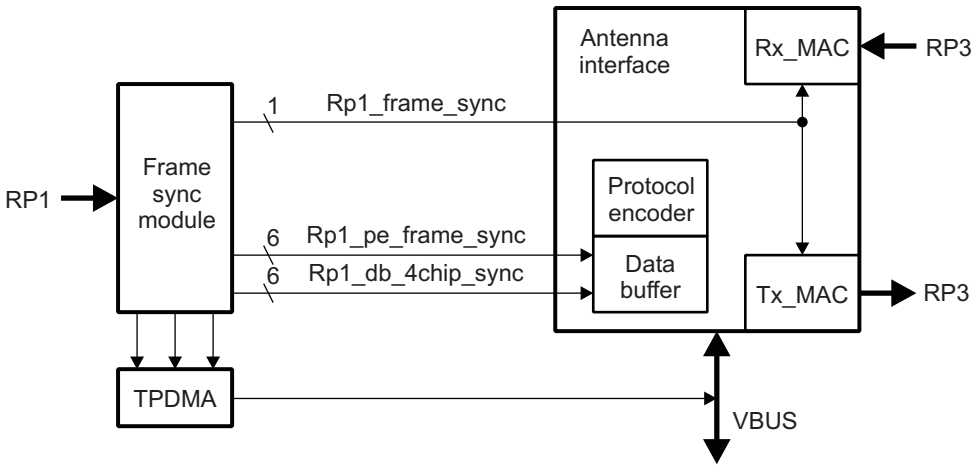
Example 10. Decombine Link 2 Into Link 0 and Link 1

```
aCdSetup.cdOutSrcSel[0] = CSL_AIF_CD_OUT_SRC_LINK_2;
aCdSetup.cdOutSrcSel[1] = CSL_AIF_CD_OUT_SRC_LINK_2;
aCdSetup.bCdEnableMask[0] = TRUE;
aCdSetup.bCdEnableMask[1] = TRUE;
aDcSetup[0].bDecombinerEnable = TRUE;
aDcSetup[0].decombinerDest[0] = CSL_AIF_LINK_0;
aDcSetup[0].decombinerDest[1] = CSL_AIF_LINK_1;
aDcSetup[0].decombinerDest[2] = CSL_AIF_LINK_0;
aDcSetup[0].decombinerDest[3] = CSL_AIF_LINK_1;
aDcSetup[0].decombinerSrcSel = CSL_AIF_LINK_2;
gblCfg.cdSetup = &cdSetup
```

Note: Combining and Decombining are specific to OBSAI and in the case of CPRI; the C/D registers should be left as is with default values. In the case of Redirection, the settings for the C/D are the same, irrespective of the protocol.

7.2 Frame Sync Module Configuration

Figure 49. Frame Sync and AIF Module System Events



The frame sync module gains frame synchronization with the external system through the OBASI RP1 or alternate frame sync interfaces. UMTS synchronization is passed to the antenna interface by way of 13 system events.

All AIF activity is synchronized to the frame sync module. TX and RX for each link have independent offset alignment to the master frame boundary. For the case of Rx MAC and Tx MAC, a single (non-offset) frame sync system event strobe provides the frame reference. The Rx MAC is programmed with six Pi offset values and the Tx MAC is programmed with six delta offset values which define the offset (relative to frame sync) for each of the six links.

For the case of PD reception of OBSAI RP3/CPRI links, no further synchronization is used since PD is a timing slave to the Rx MAC operations. Like the PD, the AG, CO, and CI are timing slaves to other modules and do not require frame sync system events for alignment.

PE operations that support transmit precede actual TM transmission and the frame sync synchronizes them separately. Here, the frame sync gives separate system events which are already offset aligned (without the use of the Pi or Delta values). For each link, the frame sync provides a frame synchronization strobe (rp1_pe_framesync) as well as additional system events that strobe every four chips in time (rp1_db_4chip_sync). Each time the four (or eight) chips of time has elapsed, the PE is enabled to process four (or eight) chips of OBSAI RP3/CPRI messages. The six fs_rp1_pe_framesync signals are intended to precede the Delta transmission timing of each link. These signals have a period of 38,400 chips. The six fs_ rp1_db_4chip_sync signals are used by the protocol encoder as a signal to commence four chips of message construction. The protocol encoder ignores these signals until the rp1_pe_framesync signal for that specific link has occurred.

In parallel, the frame sync module is also strobing system events to the TPDMA engine. These system events control the TPDMA engine to transfer data to (and from) the AIF DB. When the frame sync module is properly programmed (and VBUS is not overly loaded) the TPDMA engine completes data transfers prior to the PE message construction for this data.

In order for the AIF to properly work in conjunction with all GEM cores, peripherals, and other devices in the system, all of these frame sync system events, Pi, and Delta offset values must be programmed in a coordinated manor. For reception, the Pi offset RP1_FRAME_SYNC signal must precede the TPDMA transfer of captured data. For transmission, the TPDMA transfer must precede the PE message construction (RP1_DB_4CHIP_SYNC); The PE message construction must precede the TM message transmission (Delta offset RP1_FRAME_SYNC). For bypass or aggregation, the Rx MAC reception must precede TM transmission (Pi offset RP1_FRAME_SYNC precede Delta offset RP1_FRAME_SYNC).

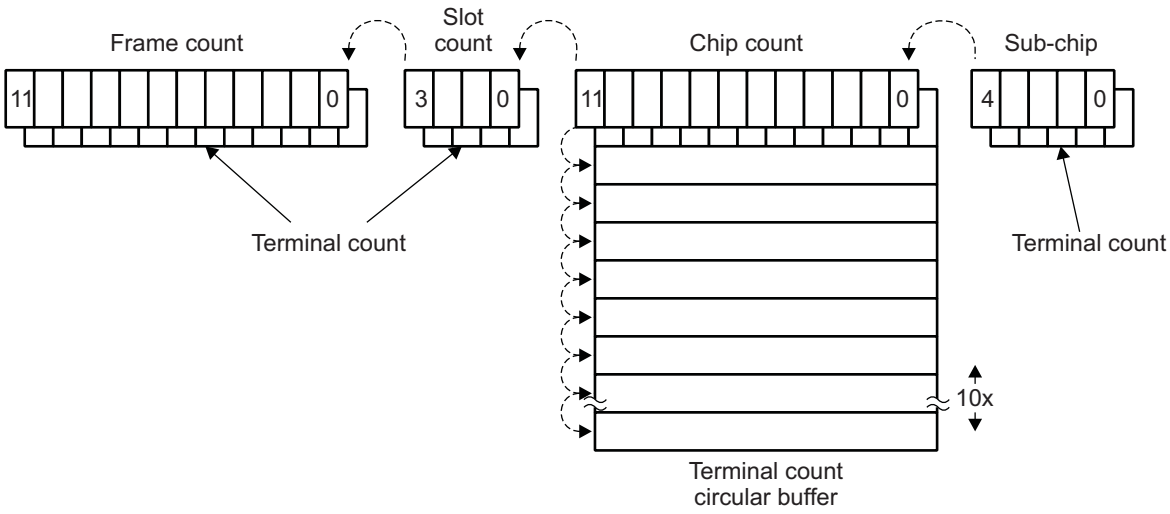
Note: For our discussion, we assume the following setup:

1. Antenna data received via the SERDES is re-transmitted via the SERDES by passing through the DSP.
2. Only one link is enabled. It is operating in 1x link rate and all of the four streams are enabled for the link.
3. Timer used: RP3 timer; Sync: UMTS Sync
4. Short frame mode is enabled; the number of chips per frame is 160 chips

7.2.1 Frame Synchronization Timer Modules, RP3 and System

System and RP3 timers are separately maintained for system and RP3 frame alignment. The RP3 timer and system timer construction is identical.

Figure 50. Timer Configuration Concept



The system and RP3 timers have four sections:

- Sub-chip (used for dividing down the input reference clock)
- Chip count
- Slot count
- Frame count

Each section is controlled by terminal counts that the DSP writes by way of VBUS. The terminal count for the chip count section is unique in that it has a circular buffer of terminal counts. The DSP must initialize the terminal count circular buffer by writing the terminal counts and the last terminal count address to control buffer depth. In the case of WCDMA FDD and RP3, the last buffer address would likely be programmed to zero.

This circular buffer serves the purpose of allowing variable chip counts to accommodate variable slot sizes within a frame (in the case of the TD-SCDMA sub-frame). Event generators may use the circular buffer state as well as the full timer value to trigger events. The system timer has its own set of terminal counts and RP3 timer has its own set. Counters will count up until they reach their respective TC values and then wrap to zero. The DSP can write the initial count values via VBUS, as will be the case when using ALTFSYNCPULSE or TRT. When using the RP1 interface, the frame count values are updated by RP1 synchronization bursts and other counters are zeroed.

For Example 11, assume the following:

- The number of slots/frame = 15
- Number of chips/slot = 2560
- Given the above, the number of chips is 38400 if the slot counter counts once to its final value.
- Therefore, TC for slot counter = 14; TC for chip counter = 2559.

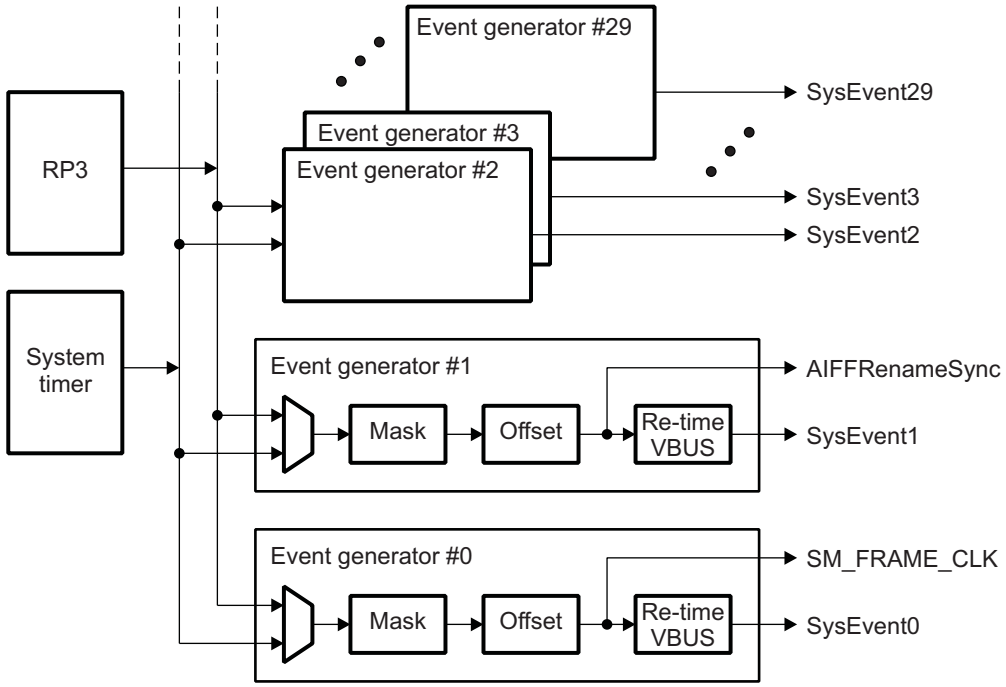
Example 11. Calculation of TC for Slot and Chip

```
/* configure terminal count */
   terminalChipCount = 2559;
   rp3TerminalCount.lastSlotNum   = 14;
   rp3TerminalCount.lastSampleNum = 31;
   rp3TerminalCount.numChipTerminalCount = 1;
   rp3TerminalCount.pLastChipNum = &terminalChipCount;

   // sys timer not used, so set all fields to zero
   sysTerminalCount.lastSlotNum   = 0;
   sysTerminalCount.lastSampleNum = 0;
   sysTerminalCount.numChipTerminalCount = 0,
   sysTerminalCount.pLastChipNum = NULL;

/* Do Config Frame sync with non-RP1 interface, in this example
   ALTFSYNCPULSE/ALTFSYNCCLK drive the Frame Sync */
   myFsyncCfg.syncRP3Timer = CSL_FSYNC_UMTS_SYNC;
   myFsyncCfg.syncSystemTimer = CSL_FSYNC_SYSTEM_TEST_SYNC;
   myFsyncCfg.clkRP3Timer = CSL_FSYNC_UMTS_CLK;
   myFsyncCfg.clkSystemTimer = CSL_FSYNC_VBUS_CLK_DIV_3;
   myFsyncCfg.pTerminalCountRP3Timer = &rp3TerminalCount;
   myFsyncCfg.pTerminalCountSystemTimer = &sysTerminalCount;
   myFsyncCfg.systemTimerRp1Sync = CSL_FSYNC_RP1_TYPE_NOT_USED;
   myFsyncCfg.rp3SyncDelay = 0;
   myFsyncCfg.systemSyncDelay = 0;
   myFsyncCfg.todSyncDelay = 0;
   myFsyncCfg.rp3EqualsSysTimer = FALSE;
   myFsyncCfg.syncMode = CSL_FSYNC_NON_RP1_SYNC_MODE;
   myFsyncCfg.reSyncMode = CSL_FSYNC_NO_AUTO_RESYNC_MODE;
   myFsyncCfg.crcUsage = CSL_FSYNC_USE_SYNC_BURST_ON_CRC_FAIL;
   // this field is don't care since Frame Sync is in non-RP1 mode
   myFsyncCfg.crcPosition = CSL_FSYNC_CRC_BIT_16_RCVD_FIRST;
   // this field is don't care since Frame Sync is in non-RP1 mode
   myFsyncCfg.todLeapUsage = CSL_FSYNC_DONT_ADD_LEAPSECS;
   // this field is don't care since Frame Sync is in non-RP1 mode

   // watchdog timer is not set since we are using non-RP1 mode
   myFsyncCfg.setupWatchDog.rp3FrameUpdateRate   = 0;
   myFsyncCfg.setupWatchDog.wcdmaFrameUpdateRate = 0;
   myFsyncCfg.setupWatchDog.todFrameUpdateRate   = 0;

   timerInit.frameNum = 0;
   timerInit.slotNum = 0;
   timerInit.chipNum = 0;

   // Init RP3 timer value
   myFsyncCfg.timerInit.pRp3TimerInit = &timerInit;
   // Init system timer value
   myFsyncCfg.timerInit.pSystemTimerInit = &timerInit;
```

7.2.2 Event Generators

There are two mechanisms for triggering and generating events. Both these types are provided for user flexibility. The first set of events is generated using masks and trigger compare value. The second set of events is generated using counters to generate an event. Each of the 30 events are either mask based (FSEVT0-9 and FSEVT18-29) or counter based (FSEVT10-17). Each one of these events can use either of the timers; that is, the system timer or RP3 timer. Figure 51 shows Frame sync events and timer connection.

Figure 51. Timer Diagram With Event Generator Detail



7.2.2.1 Masked Event Generator

Masked events provide the ability to generate UMTS trigger conditions of any 2n values of the sub-chip, chip, slot, or frame. Programmed trigger conditions are compared to either the system or RP3 UMTS timers. The trigger control consists of a trigger value register and a trigger mask register. The mask registers enable the bits that should be compared. After meeting the offset conditions, a system event generates when the enabled bits in the value register equal the enabled bits (including circular buffer state bits (chip terminal count address)) in the selected timer. The resulting trigger is periodic. These events are placed on a sub-chip granularity. You can disable these events by masking all of the bits with a write of zeros to the mask register. System events FSEVT0 through FSEVT9 and FSEVT18 through FSEVT29 are mask/trigger-based.

7.2.2.2 Counter-Based Event Generator

Counters generate the second set of events. The terminal count that loads into the counter after meeting offset (and every time the counter reaches its count-down value of zero) is programmed into the frame sync event generator value register.

A second control register, the frame sync event generator control register, selects the timer source for system or RP3, and enables or disables the timer and offset compare values. This counter has a programmable count value of period-1 when the counter reaches its full count (counts down to zero), an event generates.

The counter starts when the event is enabled and the first sync is generates from the SYNC sub-module and the offset enable is active from the event's offset counter. Offset is reached when the selected timer reaches its offset compare value.

The first event is issued when the offset value is reached. Then, the counter counts down from the terminal count value and generates another event when it reaches all zeros. The counter is reloaded with its terminal count value and repeats this process, generating periodic events based on the terminal count.

To disable each of these events, clear a programmable enable/disable bit for each of the events. System events FSEVT10 through FSEVT17 are counter based.

Table 42 gives the details about each frame sync event.

Table 42. Frame Sync Event Details

Module Entry	GEM0	GEM1	GEM2	CIC0	CIC1	CIC2	TPCC	CIC3	TIMER	AIF	RAC
FSEVT0	x	x	x					x			
FSEVT1	x	x	x					x		ai_frame_sync	
FSEVT2	x	x	x					x	x		
FSEVT3	x	x	x					x	x		
FSEVT4	x	x	x				x				lt_start
FSEVT5	x	x	x				x				
FSEVT6	x	x	x				x				
FSEVT7	x	x	x				x				
FSEVT8	x	x	x				x				
FSEVT9	x	x	x				x				
FSEVT10	x	x	x				x				
FSEVT11	x	x	x				x				
FSEVT12	x	x	x				x				
FSEVT13	x	x	x				x				
FSEVT14	x	x	x				x	x			
FSEVT15	x	x	x					x			
FSEVT16	x	x	x					x			
FSEVT17	x	x	x					x			
FSEVT18				x	x	x		x		rp1_db_4chip_sync[0]	
FSEVT19				x	x	x		x		rp1_db_4chip_sync[1]	
FSEVT20				x	x	x		x		rp1_db_4chip_sync[2]	
FSEVT21				x	x	x		x		rp1_db_4chip_sync[3]	
FSEVT22				x	x	x		x		rp1_db_4chip_sync[4]	
FSEVT23				x	x	x		x		rp1_db_4chip_sync[5]	
FSEVT24				x	x	x		x			
FSEVT25				x	x	x		x			
FSEVT26				x	x	x		x			
FSEVT27				x	x	x		x			
FSEVT28				x	x	x		x			
FSEVT29				x	x	x		x			
FS_ERR_Alarm0				x	x	x					
FS_ERR_Alarm1				x	x	x					
FS_AifFrameSync				x	x	x				rp1_frame_sync	

7.2.3 Setup Frame Sync Tick to AIF

The AIF Rx MAC and Tx MAC use the frame sync tick to synchronize the Rx/Tx SERDES links, which occurs once every frame. To configure this event, all the mask bits for sub-chip, chip, and slot should be enabled. The compare value and the offset value should be programmed if you intend to generate the trigger at a different instant than the externally provided frame sync. Otherwise, the compare and offset values for sub-chip, chip, slot and frame should be equal to zero. After configuring the mask, compare and offset values, the event should be enabled.

For Example 12 assume the following:

- Strobe period: 38400 chips
- Since the terminal count for slot and chips are already configured for 38400 chips, you only need to enable the mask bits.

Example 12. All Mask Bits Enabled to Count Until Terminal Count Value

```
//configure mask based trigger event to occur every frame, event for AIF frame sync
configMaskTrigger[0].timerUsed = CSL_FSYNC_RP3_TIMER;
configMaskTrigger[0].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_1;
configMaskTrigger[0].mask.frameMask = 0x0;
configMaskTrigger[0].mask.slotMask = 0xFF;
configMaskTrigger[0].mask.chipTerminalCountIndexMask = 0x0;
configMaskTrigger[0].mask.chipMask = 0xFFFF;
configMaskTrigger[0].mask.sampleMask = 0xFF;

configMaskTrigger[0].offset.frameOffset = 0; // don't care field
configMaskTrigger[0].offset.slotOffset = 0;
configMaskTrigger[0].offset.chipTerminalCountIndex = 0x0;
configMaskTrigger[0].offset.chipOffset = 0;
configMaskTrigger[0].offset.sampleOffset = 0;

configMaskTrigger[0].compareValue.slotValue = 0;
configMaskTrigger[0].compareValue.chipValue = 0;
configMaskTrigger[0].compareValue.sampleValue = 0;
```

7.2.4 Set up 12 Events to the AIF

The frame sync is programmed to generate strobes to the AIF every four chips for a maximum of six links. In the case of an outbound transfer, the protocol encoder uses these four chip strobes to mark the completion of EDMA operation from Tx DSP to AIF. The frame sync module provides 6 separate events to the AIF for this purpose. These are FSEVT18-FSEVT23 (within the AIF module, it is rp1_db_4chip_sync [0-5]). There is also a frame preparation tick every 38,400 chips for a maximum of six links. The frame sync module provides FSEVT24-FSEVT29 (within AIF module, it is rp1_pe_frame_sync [0-5]) for this purpose. The protocol encoder uses these frame sync prep strobes to mark the beginning of a frame. In the case of an inbound transfer, the EDMA uses the four-chip strobes to transfer data from the AIF inbound data buffer RAM to the circular DSP buffer.

Per our assumption, only one link is enabled; hence only one link is configured. If all six links are enabled, then triggers 18-23 are configured.

For Example 13 assume the following:

- Strobe period: 4 chips
- Zero out the slot mask value.
- Therefore, the slot counter value does not have any affect.
- The chip mask value is three.
- After every four counts of the chip counter, the strobe triggers. After meeting offset(1) and the compare value (10), the system fires.

Example 13. Code to Set Up Events to the AIF, Calculation 1

```
/** configure mask based trigger event generator 18 to occur every 4 chips for AIF
    * events occurs 12,16, .. */
   // event gen 18 Configuration
   configMaskTrigger[2].timerUsed = CSL_FSYNC_RP3_TIMER;
   configMaskTrigger[2].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_18;
   configMaskTrigger[2].mask.frameMask = 0;
   configMaskTrigger[2].mask.slotMask = 0;
   configMaskTrigger[2].mask.chipTerminalCountIndexMask = 0x0;
   configMaskTrigger[2].mask.chipMask = 0x3;
   configMaskTrigger[2].mask.sampleMask = 0xFF;

   configMaskTrigger[2].offset.frameOffset = 0;
   configMaskTrigger[2].offset.slotOffset = 0;
   configMaskTrigger[2].offset.chipTerminalCountIndex = 0;
```

Example 13. Code to Set Up Events to the AIF, Calculation 1 (continued)

```
    configMaskTrigger[2].offset.chipOffset = 0;
    configMaskTrigger[2].offset.sampleOffset = 0;

    configMaskTrigger[2].compareValue.frameValue = 0;
    configMaskTrigger[2].compareValue.slotValue = 0x0;
    configMaskTrigger[2].compareValue.chipValue = 1;
    configMaskTrigger[2].compareValue.sampleValue = 0;
```

For Example 14 assume the following:

- Strobe period: 38400 chips
- Since the terminal counts for slots and chips are already configured for 38400 chips, you only need to enable the mask bits.
- Event 24 alone is configured since only link is configured. If all six links are enabled, configure events 24-32.

Example 14. Code to Set Up Events to the AIF, Calculation 2

```
/** configure mask based trigger event gen 24 to occur every 38400 chips for AIF */
    //event gen 24 Configuration
    configMaskTrigger[3].timerUsed = CSL_FSYNC_RP3_TIMER;
    configMaskTrigger[3].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_24;
    configMaskTrigger[3].mask.frameMask = 0;
    configMaskTrigger[3].mask.slotMask = 0xFF;
    configMaskTrigger[3].mask.chipTerminalCountIndexMask = 0;
    configMaskTrigger[3].mask.chipMask = 0xFFFF;
    configMaskTrigger[3].mask.sampleMask = 0xFF;

    configMaskTrigger[3].offset.slotOffset = 0;
    configMaskTrigger[3].offset.chipTerminalCountIndex = 0;
    configMaskTrigger[3].offset.chipOffset = 0;
    configMaskTrigger[3].offset.sampleOffset = 0;

    configMaskTrigger[3].compareValue.slotValue = 0;
    configMaskTrigger[3].compareValue.chipTerminalCountIndexValue = 0;
    configMaskTrigger[3].compareValue.chipValue = 2;
    configMaskTrigger[3].compareValue.sampleValue = 0;
```

For timing calculation of rp1_pe_framesync or FSEVT24-FSEVT29, see Section A.2.

7.2.5 Setup EDMA Events

The EDMA data transfer from the AIF inbound data buffer RAM to the circular DSP buffer, and the EDMA transfer from the DSP circular buffer to the AIF outbound data buffer RAM are triggered by frame sync triggers. Frame sync triggers 4 to 14 are mapped to EDMA.

Frame sync triggers 4-14 are mapped to EDMA.

The periodicity of the event in Example 15 depends on the data type of the receive link being UL or DL. If it is configured as UL, then the periodicity will be 8 and the chip mask must be programmed to be 7. For calculations regarding the timing of the EDMA transfer event, see Section A.3.

For Example 15 assume the following:

- Strobe period: 4 chips
- Zero the slot mask and program the chip mask as three.

Example 15. Trigger for EDMA Transfer From AIF Inbound RAM to UL DSP

```
/** configure mask based trigger event to occur every 4 chips for EDMA x'fer (AIF
inbound RAM -> UL DSP)*/
   configMaskTrigger[1].timerUsed = CSL_FSYNC_RP3_TIMER;
   configMaskTrigger[1].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_4;
   configMaskTrigger[1].mask.frameMask = 0;
   configMaskTrigger[1].mask.slotMask = 0x0;
   configMaskTrigger[1].mask.chipTerminalCountIndexMask = 0;
   configMaskTrigger[1].mask.chipMask = 3;
   configMaskTrigger[1].mask.sampleMask = 0xFF;

   configMaskTrigger[1].offset.slotOffset = 1;
   configMaskTrigger[1].offset.chipTerminalCountIndex = 0;
   configMaskTrigger[1].offset.chipOffset = 1;
   configMaskTrigger[1].offset.sampleOffset = 0;

   configMaskTrigger[1].compareValue.frameValue = 0x0;
   configMaskTrigger[1].compareValue.slotValue = 0x0;
   configMaskTrigger[1].compareValue.chipTerminalCountIndexValue = 0;  // don't care
   configMaskTrigger[1].compareValue.chipValue = 1;
   configMaskTrigger[1].compareValue.sampleValue = 0;
```

For Example 16 assume the following:

- Strobe period: 4 chips
- Zero the slot mask and program the chip mask as three.

Example 16. Trigger for EDMA Transfer From DSP to AIF Outbound RAM

```
// event gen 5
   /** configure mask based trigger event to occur every 4 chips for EDMA x'fer (DSP ->
AIF outbound RAM) */
   configMaskTrigger[14].timerUsed = CSL_FSYNC_RP3_TIMER;
   configMaskTrigger[14].eventGenUsed = CSL_FSYNC_TRIGGER_GEN_5;
   configMaskTrigger[14].mask.frameMask = 0;
   configMaskTrigger[14].mask.slotMask = 0;
   configMaskTrigger[14].mask.chipTerminalCountIndexMask = 0;
   configMaskTrigger[14].mask.chipMask = 0x3;
   configMaskTrigger[14].mask.sampleMask = 0xFF;

   // lag of 8 chips between AIF write and EDMA read
   configMaskTrigger[14].offset.slotOffset = 1;
   configMaskTrigger[14].offset.chipTerminalCountIndex = 0;
   configMaskTrigger[14].offset.chipOffset = 1;
   configMaskTrigger[14].offset.sampleOffset = 0;

   configMaskTrigger[14].compareValue.slotValue = 0;
   configMaskTrigger[14].compareValue.chipValue = 9;
   configMaskTrigger[14].compareValue.sampleValue = 0;
```

Note: FSEVT4 is also routed to RAC and signifies the start of the RAC operation. So if RAC is operational this event should not be used for any other purposes and the event should be configured to generate an event every 32 chips. In that case, the slot mask and frame mask should be 0, chip mask should be 0x1f, and sub-chip mask should be 0xff.

7.3 Achieving Link Synchronization

7.3.1 OBSAI RP3 Receiver Synchronization

The OBSAI RP3 receiver state machine consists of four states: UNSYNC, WAIT_FOR_K28.7_IDLES, WAIT_FOR_FRAME_SYNC_T, and FRAME_SYNC. Two of these states, namely WAIT_FOR_K28.7_IDLES and WAIT_FOR_FRAME_SYNC_T, can be considered to form a single logical state called SYNC.

The states UNSYNC, SYNC, and FRAME_SYNC are defined as:

- UNSYNC: bus link is down. A lot of byte errors are detected.
- SYNC: bus link is working (i.e., a connection exists).
- FRAME_SYNC: normal operational mode. Frame structure is detected and messages are received.

The receiver state machine uses two separate criteria to determine the quality of a bus link; the first one monitors the byte error rate of the link while the second one monitors the validity of the received frame structure. Parameters BLOCK_SIZE, SYNC_T, UNSYNC_T, FRAME_SYNC_T, and FRAME_UNSYNC_T control the state transitions. The parameters SYNC_T, UNSYNC_T, FRAME_SYNC_T, and FRAME_UNSYNC_T are programmable at a global level. On reset, the state machine enters the state UNSYNC. State transition to WAIT_FOR_K28.7_IDLES is done if SYNC_T consecutive blocks of bytes have been properly received. BLOCK_SIZE bytes in each block, and a block is valid if all of the bytes were received correctly (no 8b10b decoding errors). Otherwise, the block is invalid. This measurement is made without regard to K character boundaries.

The transition from state WAIT_FOR_K28.7_IDLES back to UNSYNC is done if UNSYNC_T consecutive invalid byte blocks are received or in case the hardware resets.

The master frame boundary is indicated by a K28.7 IDLE byte. The transition from WAIT_FOR_K28.7_IDLES to WAIT_FOR_FRAME_SYNC_T is done when a K28.7 IDLE byte (i.e., a possible master frame boundary), is detected. In the WAIT_FOR_FRAME_SYNC_T and the FRAME_SYNC states, master frame timing is fixed (defined by the received K28.7 IDLE byte).

The validity of consecutive message groups is observed in the WAIT_FOR_FRAME_SYNC_T state. The FRAME_SYNC is entered when consecutive valid messages groups are received. The counters that provide the predicted K character indicators are trained in this state. If the FRAME_UNSYNC_T consecutive invalid message groups are received, the WAIT_FOR_K28.7_IDLES state is entered and a search for a new K28.7 IDLE starts immediately.

The K character indicators that feed the PD and CD are created by counters that were trained to predict when to provide these indicators in FRAME_SYNC state. If the WAIT_FOR_FRAME_SYNC_T state is re-entered, the counters go back into training.

The state transition from FRAME_SYNC to WAIT_FOR_K28.7_IDLES occurs when FRAME_UNSYNC_T consecutive invalid message groups are received. The transition from FRAME_SYNC to UNSYNC occurs when UNSYNC_T consecutive invalid blocks of bytes are received.

A valid message group is a block of 400 bytes where the first 399 bytes are of type data or 8b10b decoding error. Idle codes K28.5 or K28.7 are prohibited. The last IDLE byte in a block must be either K28.5 or K28.7 idle bytes or 8b10b decoding errors.

Furthermore, the order of the idle bytes matters. In the first N_MG-1 message groups of a master frame, N_MG-1 idle bytes of the message group must be equal to K28.5, while the IDLE code shall be K28.7 in the last message group of the master frame.

The FRAME_UNSYNC_T COUNTER counts the missing or extra K28.x bytes and clears the FRAME_SYNC_T counter. A “missing” K28.x byte is not counted if there is an 8b10b decode error in its place.

Note: The Comma alignment should be enabled during link synchronization. Once the link is synchronized comma alignment must be disabled since K28.7 followed by data groups /D3.y/ ,/D11.y/, D12.y/, D19.y/, /D20.y/, /D28.y/ or /K28.y/, where y is an integer in the range 0 through 7 may cause the receiver to change the 10-bit code group alignment.

7.3.2 CPRI Frame Synchronization

There are four defined states: XACQ1, XACQn, XSYNCn, and HFNSYNC. The XACQ1 state is equivalent to the RP3 UNSYNC state. The bus link is down and byte errors are being detected in the XACQ1 state. This is the state that the Rx MAC comes up in after reset and when LOS is detected. While in the XACQ1 state, the RM_LOF_STATE is set to true (1). The synchronization counters clear at the transition from state ST0 to ST1.

When the idle byte K28.5 is detected and LOS = 0, the XACQn state is entered. If the Idle byte is not detected and the Y_CNTR, W_CNTR and X_CNTR SYNC counters have all counted back to zero, the state machine reverts back to the XACQ1 state. However, if idle bytes are detected at the correct time, the state machine transitions to the XSYNCn state. The number of correct idle byte detections before proceeding to the next state is programmable via the SYNC_T COUNTER. The RM_LOF_STATE is set to true (1) when it is in the XACQn state.

If an idle byte is not detected and if the Y_CNTR, W_CNTR and X_CNTR SYNC counters have all counted back to zero while in the XSYNCn state, the state machine reverts back to the XACQ1 state. The number of incorrect idle bytes before making this transition is programmable via the UNSYNC_T COUNTER. However, if idle bytes are detected at the correct time, the state machine transitions to the HFNSYNC state. The number of correct idle byte detections before proceeding to this state is programmable via the FRAME_SYNC_T counter. The LOF is set to false (0) when in the XSYNCn state.

The state machine remains in the HFNSYNC state as long as idle bytes are detected at the correct time. However, if idle bytes are not detected at the correct time, the state machine transitions to the XSYNCn state. The number of incorrect idle byte detections before reverting to the XSYNCn state is programmable via the FRAME_UNSYNC_T counter. LOF is set to false (0) while in the HFNSYNC state.

The following comprise the CPRI synchronization counters:

- Y_CNTR = byte number within a word (0, 0 to 1, or 0 to 3, depending on link rate 1x, 2x, or 4x, respectively)
- W_CNTR = word counter in a basic frame (0 to 15)
- X_CNTR = basic frame counter (0 to 255)

7.3.3 Reception Synchronization

There are six Rx MACs that each independently check incoming SERDES data for synchronization. Once operational, the 8b10b decoder passes the Rx MACs synchronization K characters indicating the frame synchronization embedded in the SERDES data stream. The Rx MAC checks the alignment of this embedded frame sync against the frame sync FS_AIFFRAMESYNC_PI signal provided by the frame sync module.

A frame synchronization protocol executes under conditions where frame sync has failed (or has not been established yet). In the frame synchronization protocol, the Rx MAC zeros all of its FIFO pointers and sets the MSTR_FRAME_BNDY_OUT_OF_RANGE bit in RM_LINK_STSA register (see Section 8.3.2.1). It stays in this state until the frame sync is established.

7.3.3.1 Receiver Synchronization State Machine

The receiver synchronization state machine has several outputs. The current state of the receiver is available for the application layer and an event generates from each state change.

Table 43. RX Sync FSM Output [(1)]

State	OBSAI RP3 Output	CPRI Output
ST0	Rm loss of signal = 1	Rx MAC LOSS of signal = 1, Rx MAC LOF state = 1
ST1	n/a	RM_LOF_state = 1, Set x, y and w counters to 0 at transition from ST0 to ST1.
ST2	n/a	n/a
ST3	Rm cd rdy = 1	Rx MAC HFSYNC state =1, Rx MAC CD RDY = 1

(1) FSM output signals are inactive 0 unless shown otherwise.

Note: The Rx MAC indicates the boundary of a master frame (K28.7) for OBSAI RP3 or hyper frame (K28.5 and HFN + 149) for CPRI.

It is possible to force the receive synchronization state machine states with the RM_FORCE_RX_STATE configuration bits (see Section 8.2.5.1). Once in the forced state, the FSM remains in this state unless forced to another state, or the forcing is removed. Other than not being able to change the FSM states when forced, the Rx MAC should behave as normal for that state condition.

Table 44. RX Sync FSM State Names

State	OBSAI RP3 Name	CPRI Name
ST0	UNSYNC	XACQ1
ST1	WAIT_FOR_K28.7_IDLES	XACQn
ST2	WAIT_FOR_FRAME_SYNC_T	XSYNCn
ST3	FRAME_SYNC	HFSYNC

Figure 52. RX Synchronization State Diagram

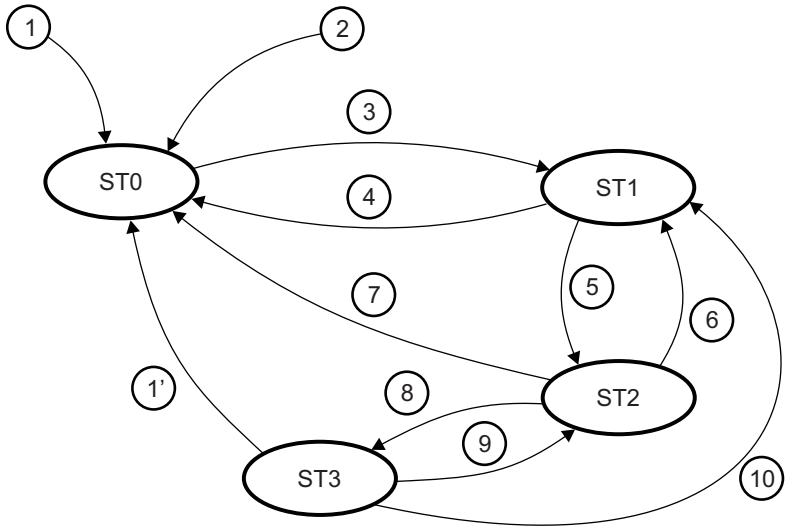


Table 45. RX Sync FSM State Transition Table

Transition	OBSAI RP3 Description	CPRI Description
1	RST_N	RST_N
2	Rx MAC new PI offset (PI_OFFSET updated) | SD_RM_LOS detect	Rx MAC new PI offset | Rx MAC loss detect (N × 8b10b errors in a hyper frame) | SD_RM_LOSS_DETECT
3	SYNC_T consecutive valid blocks of bytes received	K28.5 byte received & ~(rm loss detect | SD_RM_LOSS_DETECT)
4	UNSYNC_T consecutive invalid blocks of bytes received	K28.5 byte received & (Y_CNTR = 0 & W_CNTR = 0 & X_CNTR = 0)
5	One K28.7 idle byte received	(K28.5 byte received & (Y_CNTR = 0 & W_CNTR = 0 & X_CNTR = 0)) occurs SYNC_T consecutive times
6	FRAME_UNSYNC_T consecutive invalid message groups received	N/A
7	UNSYNC_T consecutive invalid blocks of bytes received	(K28.5 byte !received & (y_cntr = 0 & w_cntr = 0 & x_cntr = 0)) occurs UNSYNC_T consecutive times
8	FRAME_SYNC_T consecutive valid message groups received	(K28.5 byte received & (y_cntr = 0 & w_cntr = 0 & x_cntr = 0)) FRAME_SYNC_T consecutive times
9	N/A	(K28.5 byte received & (Y_CNTR = 0 & W_CNTR = 0 & X_CNTR = 0)) FRAME_UNSYNC_T consecutive times

Table 45. RX Sync FSM State Transition Table (continued)

Transition	OBSAI RP3 Description	CPRI Description
10	FRAME_UNSYNC_T consecutive invalid message groups received (Idle order matters)	N/A
11	UNSYNC_T consecutive invalid blocks of bytes received	N/A

Note: These parameters have fixed values: M_MG = 20, K_MG = 1, and BLOCK_SIZE = 400. N_MG = 1920 for a 1x link, N_MG = 3,840 for a 2x link and N_MG = 7,680 for a 4x link.

7.3.3.2 Pi Offset

The purpose of frame boundary synchronization is to minimize buffering needs in bus nodes (e.g., to ensure that corresponding messages are received at the same message slot at each bus node and taking into account slot offsets defined by the PI_OFFSET parameter). This mechanism is required for OBSAI RP3 operation, but may be useful for CPRI operation as well. In CPRI mode, only positive offsets are supported.

The synchronization function indicates the received master frame boundary for each receiver. An additional offset parameter (PI_OFFSET) is provided for each receiver that indicates the earliest possible time instant when a frame boundary can be received. This time instant, called reference time, is equal to FRAME SYNC TIC plus PI_OFFSET. An exact value of pi_offset is provided in run time mode so that the first byte of the frame boundary is received at the reference time or at maximum, a programmable window of time (RM_VALID_MSTR_FRAME_WIND) that is specified in VBUS_CLK tics. When the received master frame boundary is detected outside of this allowed window (RM_MSTR_FRAME_BNDY_OUT_OF_RANGE), an error occurs. A received master frame boundary is defined as a received K28.7 IDLE for OBSAI or (K28.5 AND HFN=149) for CPRI.

For OBSAI RP3 only, an error may not generate if a received K28.7 character is detected outside of the RM_VALID_MSTR_FRAME_WIND, but a predicted K28.7 character is not present. The RM_EXTRA_K28P7_ERR_SUPPRESS bit in the Rx MAC configuration MMR can configure this last behavior.

Table 46. Received K28.7 Detection Behavior (X = don't care)

Suppress extra k28.7 error ?	Received_k28.7 ?	Predicted_k28.7 ?	VALID_MSTR_FRAME_WIND ?	Error ?
X	N	N	X	N
X	N	Y	X	N
X	Y	Y	Y	N
X	Y	Y	N	Y
N	Y	N	X	Y
Y	Y	N	X	N

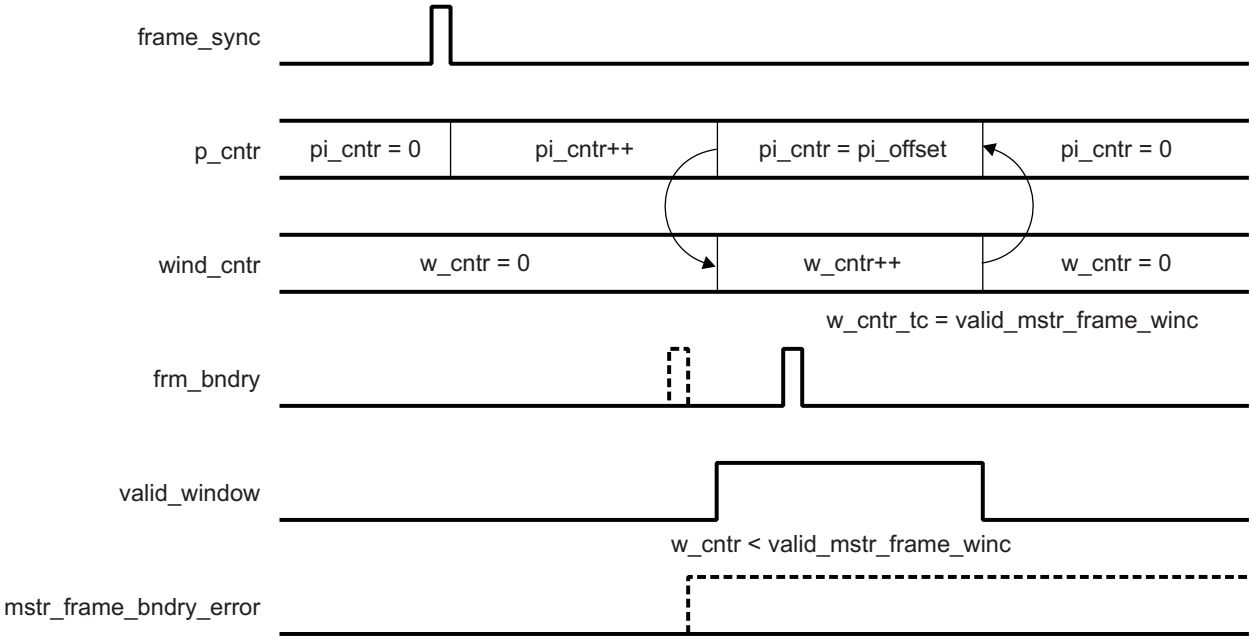
The Rx MAC can also perform measurements of the received frame boundary. The received frame boundary can be detected within a programmable time window (RM_RCVD_MSTR_FRAME_WIND) specified in the VBUS_CLK TICS after RM_PI_OFFSET. If the frame boundary offset is not detected during this window, the measurement is saturated at a programmable limit (RM_RCVD_MSTR_FRAME_WIND). The measurement is only activated when the receiver FSM is in state ST3 and the measurement is conducted once per UMTS frame.

The value of parameter PI_OFFSET is fixed at run time. Updating the parameter automatically forces the receiver state machine to state ST0. For OBSAI, the downlink direction value of PI_OFFSET is typically negative while taking a positive value in the uplink direction.

The PI_OFFSET value is a 2's complement number; thus, it can take on both positive and negative values. Since there is no such thing as negative time, to make this work the reception of master frame boundary is relative to the frame sync signal originating in the frame sync module.

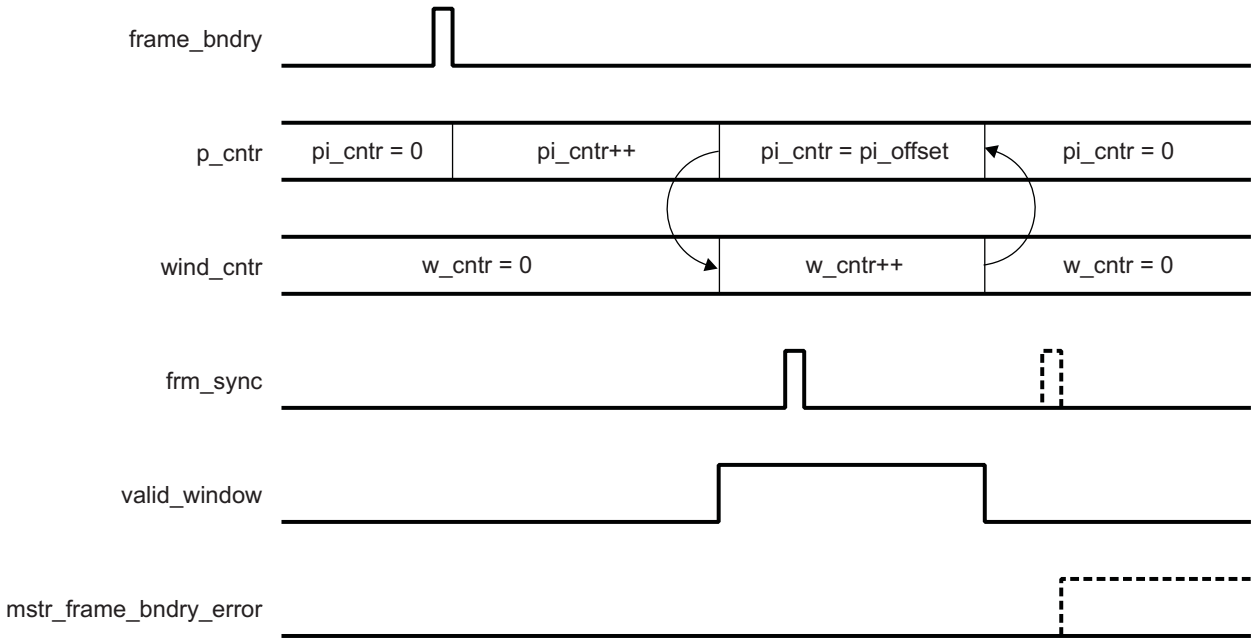
When the value of PI_OFFSET is positive, as shown in Figure 53, the reception of the frame sync starts the PI_OFFSET counter. When the PI_OFFSET counter reaches its programmed value terminal count, the VALID_WINDOW counter starts counting. When the VALID_WINDOW counter reaches its programmable terminal count, both counters clear to wait for the next frame sync. If a master frame boundary is detected anywhere outside of the valid window, as specified in Table 46, the RM_MSTR_FRAME_BNDY_OUT_OF_RANGE error is detected.

Figure 53. Pi Counter for Positive PI_OFFSET (OBSAI and CPRI)



When the value of PI_OFFSET is negative, as shown in Figure 54, the reception of the master frame boundary starts the PI_OFFSET counter. When the PI_OFFSET counter reaches its programmed value terminal count, the VALID_WINDOW counter starts counting. When the VALID_WINDOW counter reaches its programmable terminal count, both counters clear to wait for the next master frame boundary. If a frame sync is not detected in the valid window, the RM_MSTR_FRAME_BNDY_OUT_OF_RANGE error is detected. If no master frame boundary is detected, the counters do not start counting and no error is reported.

Figure 54. Pi Counter for Negative PI_OFFSET (OBSAI only)



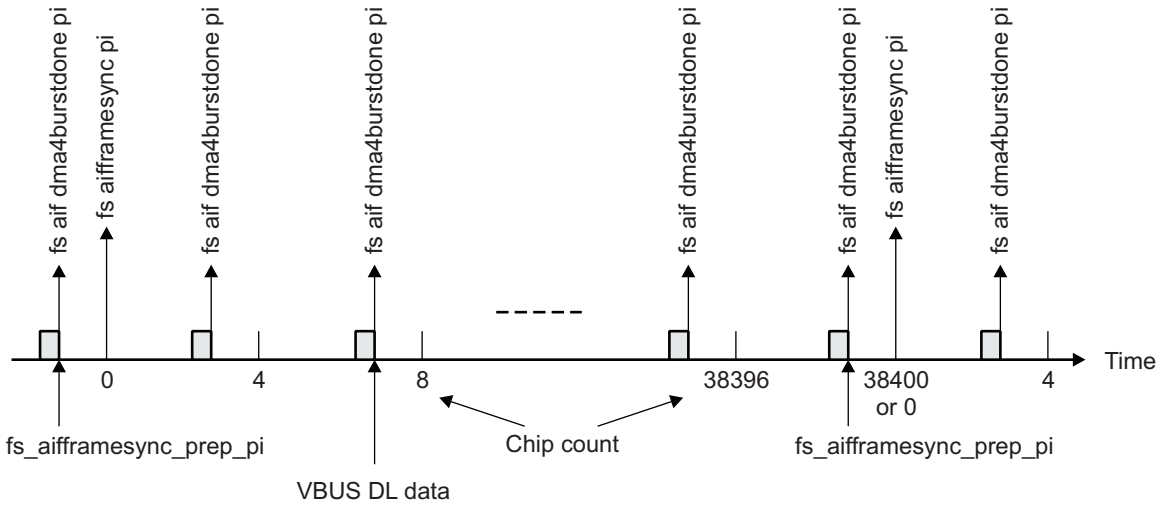
Adjustment of Pi Offset

To achieve receive synchronization, the following condition should be satisfied: time of reception of first byte of frame boundary < (frame_sync + pi_offset + rm_valid_mstr_frame_wind). You should measure the time instants above in VBUS CLOCK tics.

7.3.3.3 Hardware Components of Receiver Synchronization

The timing relationship of the frame sync signals for a DL example is shown in Figure 55. The squares represent the time it takes to transmit the DL data from the DSP core(s) to the AIF.

Figure 55. Timing Diagram of Frame Synchronization Signals for DL



The loss of received frame sync is detected by the Rx MAC and indication of the sync loss is passed down stream. Within TMS320TCI6487/8, the Rx MAC alerts the protocol decoder and combiner/decombiner that the link is down. Additionally, the Rx MAC triggers an error/alarm condition for possible interrupt intervention by a DSP core. The protocol decoder stops extracting data from the affected link. There are multiple error conditions that can cause the Rx MAC link to be down. The software can determine which error condition caused the link to be down by reading the error/alarm registers.

The combiner/decombiner compensates for down links when it can (loss of frame synchronization is considered down). When combining, the combiner continues operation if at least one of the combined input links is up. The combiner inserts empty messages into missing combined links. When passing-through, decombining, or combining where none of the combined input links is up, the combiner does not have an appropriate timing reference. Under these conditions, the combiner/decombiner stops operation and passes a control signal indicating down link to the aggregator.

When a link goes down, the aggregator uses protocol encoder data, if possible. Protocol encoder data is normally programmable and is inserted and aggregated with the RX combined/decombined data stream. In the event that a link is down and the link has protocol encoder contributions, the aggregator inserts the protocol encoder link. In the event that the link does not receive protocol encoder contributions, the aggregator indicates that the link is down to the Tx MAC.

When a Tx MAC is starved of received data (either due to a non-functional link or some other error condition) it flags an error condition, constructs / transmits empty messages, and inserts appropriate K characters. Misaligned or missing link data to the Tx MAC is a data synchronization error.

In the case of OBSAI, reception synchronization fails if a single K28.7 character or two consecutive K28.5 characters are missed in the RX SERDES data stream. A K character is considered to be missed if it fails to arrive within five 76.8 MHz clock windows of the Pi value. Reception synchronization recovers on the next K28.7 character, which arrives within RM_RCVD_MSTR_FRAME_WIND after Pi offset.

Note: SERDES links have bit error rates (BER) with approximate frequencies of one error per day. A single SERDES bit error only causes frame sync failure if it affects the K28.7 character, or two consecutive K28.5 characters. The probability of this occurrence is on the order of once per 400 years (per link).

7.3.3.4 Software Components of Receiver Synchronization

Rx MAC must configured with the parameters discussed in Section 7.3.3.3. For example:

```
// populate Rx MAC link fields
aRmCfg[0].bEnableRxMac = TRUE;
aRmCfg[0].maxMasterFrameOffset = 200;
aRmCfg[0].piOffset = 700;
aRmCfg[0].validMasterFrameOffset = 200;
```

7.3.4 Transmission Synchronization

The Tx MAC creates the frame structure based on the programmed link rate of 1x, 2x, or 4x speed links (programmed through the LINK_RATE field in the LCFG register). This includes the insertion of K28.7 (master frame boundary) and K28.5 IDLE bytes (message group boundary) for OBSAI RP3. It also includes the insertion of K28.5 comma bytes for a 1x rate link plus D16.2 and D5.6 bytes for 2x and 4x rate links for CPRI. The frame begins transmitting with an offset in time based on the value programmed for DELTA OFFSET. Frame transmission begins when the transmit state machine reaches the FRAME_TX state, as illustrated below.

The received frame structure is compared to the Tx MAC generated frame structure, based on an idle byte as the frame delimiter from the AG (via the CO). If a miss compare results, the DATA_NOT_ALIGNED bit in the TM_LINK_STS register (see Section 8.3.3) is set.

7.3.4.1 Transmitter Synchronization State Machine

There are three states in the transmitter state machine: OFF, IDLE, and FRAME_TX. The state machine enters the initial OFF state on reset. Transmission is disabled and nothing is transmitted in this state.

The application layer controls the transition from the OFF state to the IDLE state. The state is changed when the application layer sets the configuration parameter TM_TRANSMITTER_EN equal to 1 and one of the following cases is true:

- Parameter TM_LOS_EN is set equal to zero, meaning that signal RM_LOSS_OF_SIGNAL (see Section 8.3.2.1) from the receiver state machine has no impact on the transmitter state.

- TM_LOS_EN is equal to 1 and RM_LOSS_OF_SIGNAL is equal to 0 (inactive).

In the IDLE state, the transmitter continuously transmits K28.5 IDLE bytes, based on which receiver end can obtain sample (byte) synchronization. The transmitter state machine always remains in the IDLE state at least *t* micro seconds via the TM_FRAME_XMIT_EN configuration register bit. The value of *t* must be large enough to allow the phase locked loop (PLL) of the SERDES macro to settle and the interfacing receiver to obtain correct sample phase. Assuming that the ai_ref_clk is 122.88 MHz, the *t* value is around 2.66 s. (*t* = 1 s + 200 SERDES ref clock. For details refer to *TMS320TCI6487/8 SERDES Hardware Design Guidelines*.

Transition from the IDLE state to the FRAME_TX state occurs when the application layer updates (modifies) the value of parameter DELTA_OFFSET.

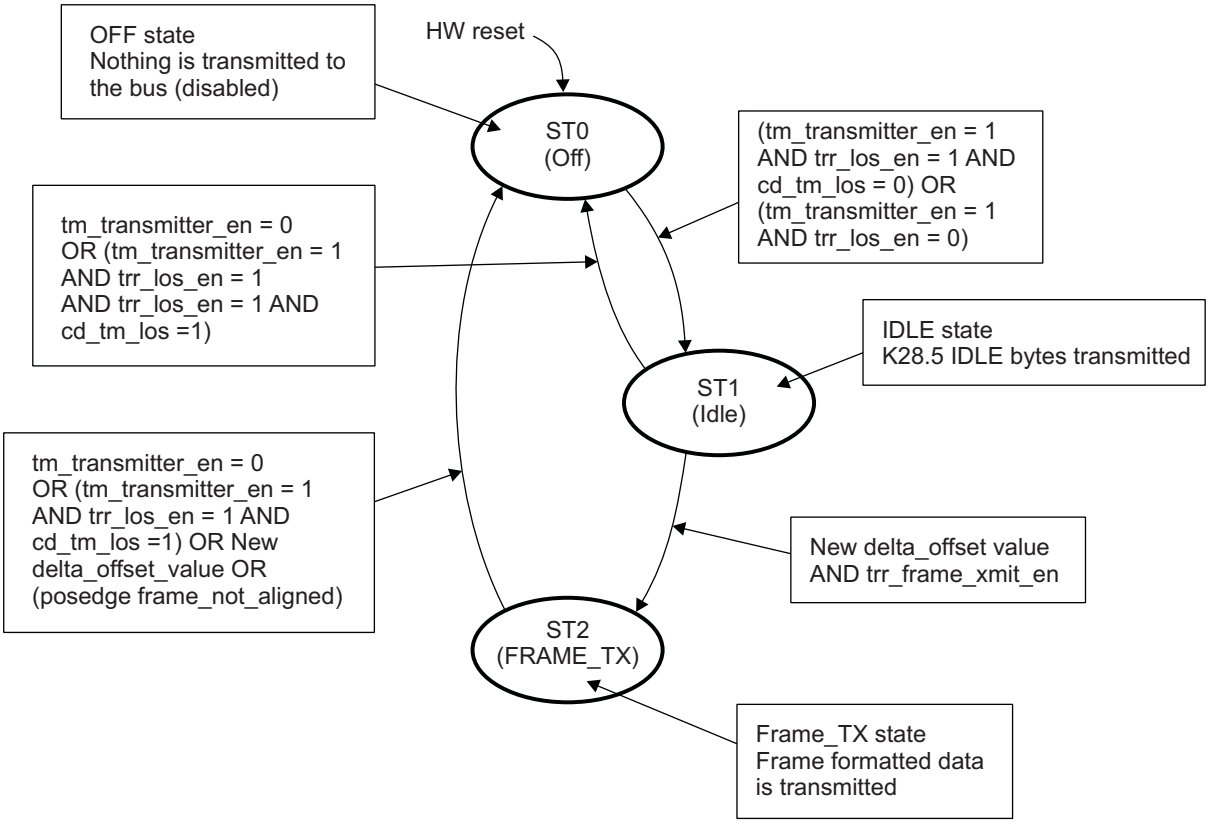
In the FRAME_TX state, transmission of the valid frame structure is performed within 20ms of transitioning from the IDLE state. Valid messages as well as RP3 empty messages are transmitted (in the form of all data bytes = 0xFF) while in the FRAME_TX state. In the case of CPRI, synchronization and L1 inband control messages are transmitted as well as NULL data bytes when there is no other data to transmit.

Writing a new DELTA_OFFSET value transitions the state machine back into the OFF state. The OFF state is also entered when the frame alignment error is detected (see Section 7.3.4.2). A reset or active LOSS_OF_SIGNAL from the receiver state machine may force the state to OFF (transmission disabled).

The transmit synchronization state machine states can be forced through the TM_FORCE_TX_STATE configuration register bits.

(See Section 8.2.6.1 for information on the TM_TRANSMITTER_EN, TM_LOS_EN, TM_FRAME_XMIT_EN, and TM_FORCE_TX_STATE bits of the TM_LINK_CFGA register.)

Figure 56. Transmit Synchronization State Diagram

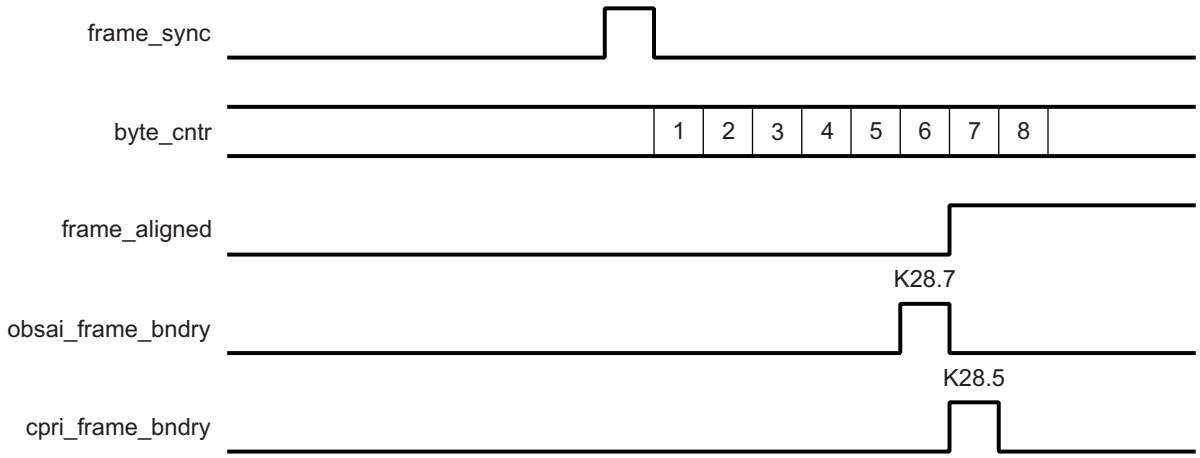


7.3.4.2 Delta Offset

The transmission time of the start of frame is synchronized to the frame sync signal generated by the frame sync module located external to the AIF. The first byte of the frame (idle byte) is transmitted at DELTA_OFFSET time from the frame sync signal provided by the frame sync module. This mechanism is required for OBSAI RP3 operation, but may be useful with CPRI operation as well. Only positive offsets are supported in CPRI mode. A specific DELTA_OFFSET parameter value is used for each transmitter port and is fixed at run time.

After the first frame sync pulse has been detected, the frame byte counter begins counting byte clocks up to the maximum number of byte clocks that is operating mode and link-rate dependent (see Figure 57). Frame alignment logic then monitors the relative position of frame sync to the byte count. Due to synchronization across clock boundaries, the frame sync is allowed to dither by one byte clock in either direction.

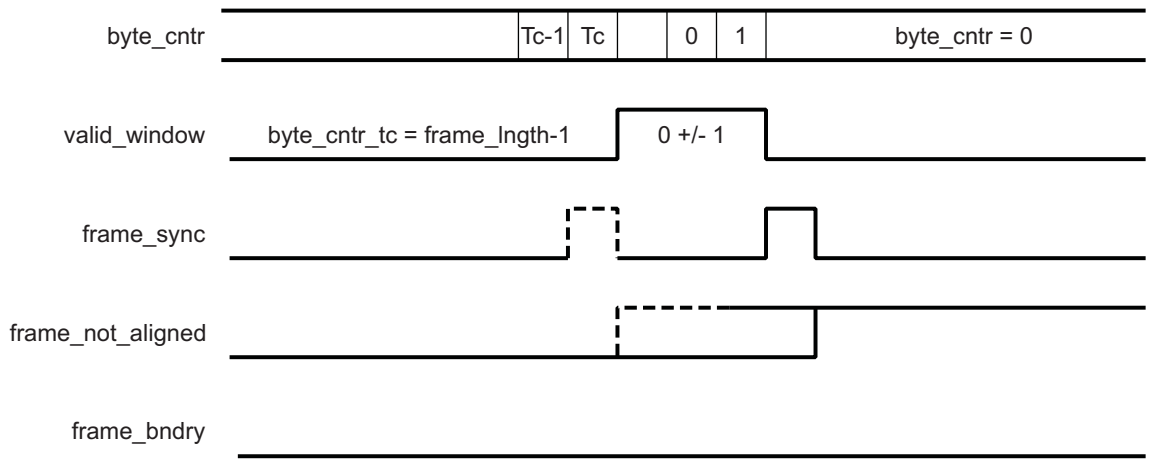
Figure 57. Transmit Frame Sync Timing (DELTA_OFFSET = 6)



If the alignment of the frame boundary and frame sync differ by more than +/- one byte clock, a frame alignment error (TM_FRAME_NOT_ALIGNED) is flagged (see Figure 58) and the FSM transitions to the OFF state (ST0). If the transmitter is still enabled, the FSM then transitions to the IDLE state (ST1) and waits for a new DELTA_OFFSET value to be programmed. The HFN counter is cleared when the frame alignment error is detected..

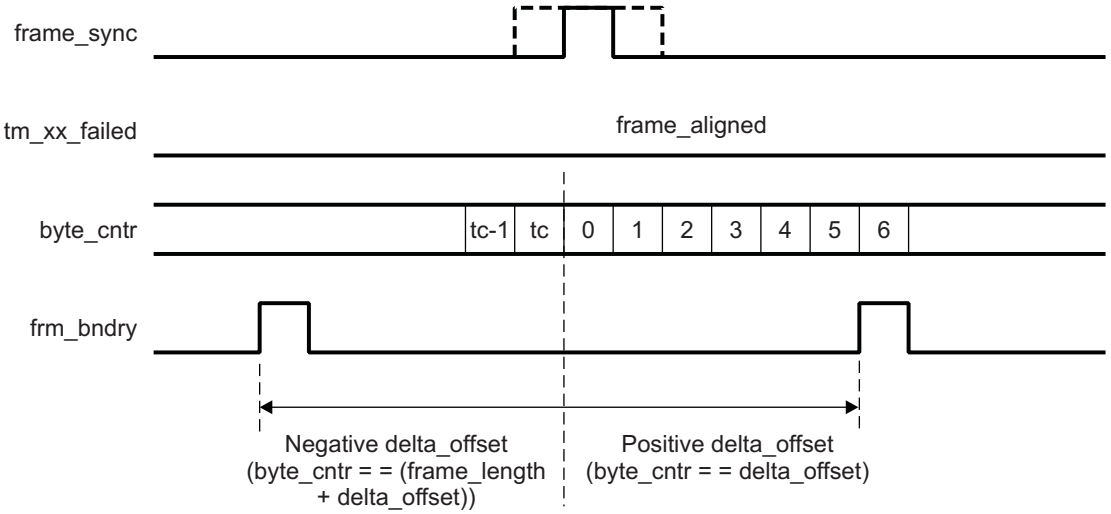
When frame alignment fails, upstream modules such as the CD and PE stop sending the data until the start of the next frame boundary. When frame alignment is once again achieved, the Tx MAC continues transmitting empty messages in the case of OBSAI RP3 or, in the CPRI case, the basic frame structure is transmitted with only synchronization and L1 inband signaling, with all other fields being zero. This continues until valid data is available for transmission. Once the data is available to the TM, it will start transmitting data when frame synchronization is achieved and DELTA_OFFSET time has elapsed.

Figure 58. Transmit Frame Sync Error Timing



During normal operation (see Figure 59), after synchronization has been achieved, the frame byte counter (BYTE_CNTR) counts continuously from 0 to the terminal count (BYTE_CNTR_TC).

Figure 59. Transmit Normal Operation Timing (DELTA_OFFSET = 6)



The terminal count value for the byte counter is the frame length (FRAME_LNGTH) minus 1. The frame length is determined by the OP_MODE (i.e., OBSAI or CPRI) and the link rate as shown in Table 47.

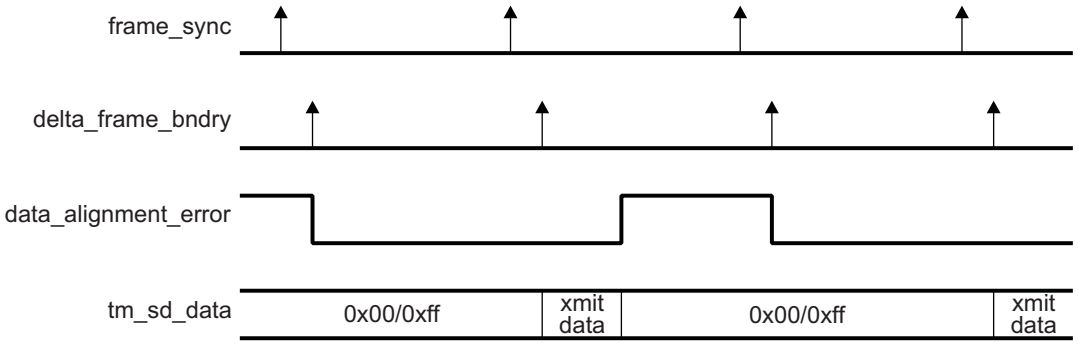
Table 47. Frame Length vs. Link Rate

Link Rate	OBSAI RP3 Frame Length (BYTE_CLOCK TICKS)	CPRI Frame Length (BYTE CLOCK_TICKS)
1x	768,000	614,400
2x	1,536,000	1,228,800
4x	3,072,000	2,457,600

Data alignment between the incoming data stream and the TM is checked through the position of the idle bytes. The incoming Idle bytes are verified against the generated idle bytes. The generated idle bytes are the actual idle bytes that are transmitted. If a miss compare occurs, the data not aligned error (TM_DATA_NOT_ALIGNED) is flagged. At this point, the Tx MAC FIFO pointers clear and the empty messages (in the form of all data bytes = 0xFF) are transmit; or in the case of CPRI, zeros (NULL bytes) are inserted.

When frame alignment has failed, upstream modules such as the CD and PE stop sending the data until the start of the next frame boundary. When frame alignment is once again achieved, the TM continues transmitting empty messages in the case of OBSAI RP3 or, in the CPRI case, the basic frame structure is transmitted with only synchronization and L1 inband signaling with all other fields being zero. This continues until valid data is available to transmit. Once the data is available to the TM, it starts transmitting data when frame synchronization is achieved and DELTA_OFFSET time has elapsed (see Figure 60).

Figure 60. Data Not Aligned Operation



If the TM was outputting data and its FIFO then runs dry, it immediately outputs empty messages until the FIFO has data in it and the next DELTA_OFFSET frame boundary is reached. The TM pulses the TM_XX_FAILED signal at each frame boundary, if the FIFO remains empty.

7.3.4.3 Hardware components of Transmit synchronization

The Tx MAC state machines dictate a form of data synchronization as well as frame synchronization. Embedded in the data streams are K characters (OBSAI/CPRI) and D characters (CPRI only); these characters are expected at precise locations within the data. The state machines within the Tx MAC dictate the expected location of these characters. As the Tx MAC reads data from its input FIFO, it checks the alignment of these characters. If the character alignment fails, the data synchronization error condition is activates and the data re-synchronization protocol is executes. There is no hardware support to determine the last piece of data that was correctly transmitted.

Frame Re-Synchronization Protocol

The Tx MAC automatically initiates data re-synchronization protocol in conjunction with the frame re-synchronization operation. Once frame synchronization is lost, the Tx MAC indicates that it is not ready for data to the rest of the AIF. All portions of AIF that source data to the Tx MAC flush their respective FIFOs and stop gathering/generating data for transmission. The Tx MAC re-synchronizes its frame rate state machines to the frame sync strobes.

Data Re-Synchronization Protocol

The Tx MAC zeros read and write pointers to its FIFO, essentially flushing all FIFO contents. The Tx MAC indicates to all other portions of the AIF that the link is not ready for transmission data. All FIFOs within the AIF that feed this particular link are flushed (in this case, flushing simply zeros read and write pointers; memory locations do not actually clear). For transmission, the generation of data is stopped.

The software is informed of this condition via an error/alarm event. The software can continue its synchronization with the outbound RAM up to a point. If the length of time for re-synchronization is greater than the amount of time it takes to fill the allocated RAM; then the new data overwrites the old data.

At the start of the next master frame boundary, the Tx MAC is ready to restart data synchronization and all blocks feeding this transmission link are notified. Then, all bocks that gather or generate data for this link start feeding FIFOs at the beginning of the next frame boundary.

7.3.4.4 Software Components of Transmit Synchronization

The delta parameter discussed in the above sections should be programmed in the Tx MAC module. For example:

```
// populate Tx MAC link fields
   aTmCfg[0].bEnableTxMac = TRUE;
   aTmCfg[0].bEnableFrameXmit = TRUE;
   aTmCfg[0].deltaOffset = 570;
```

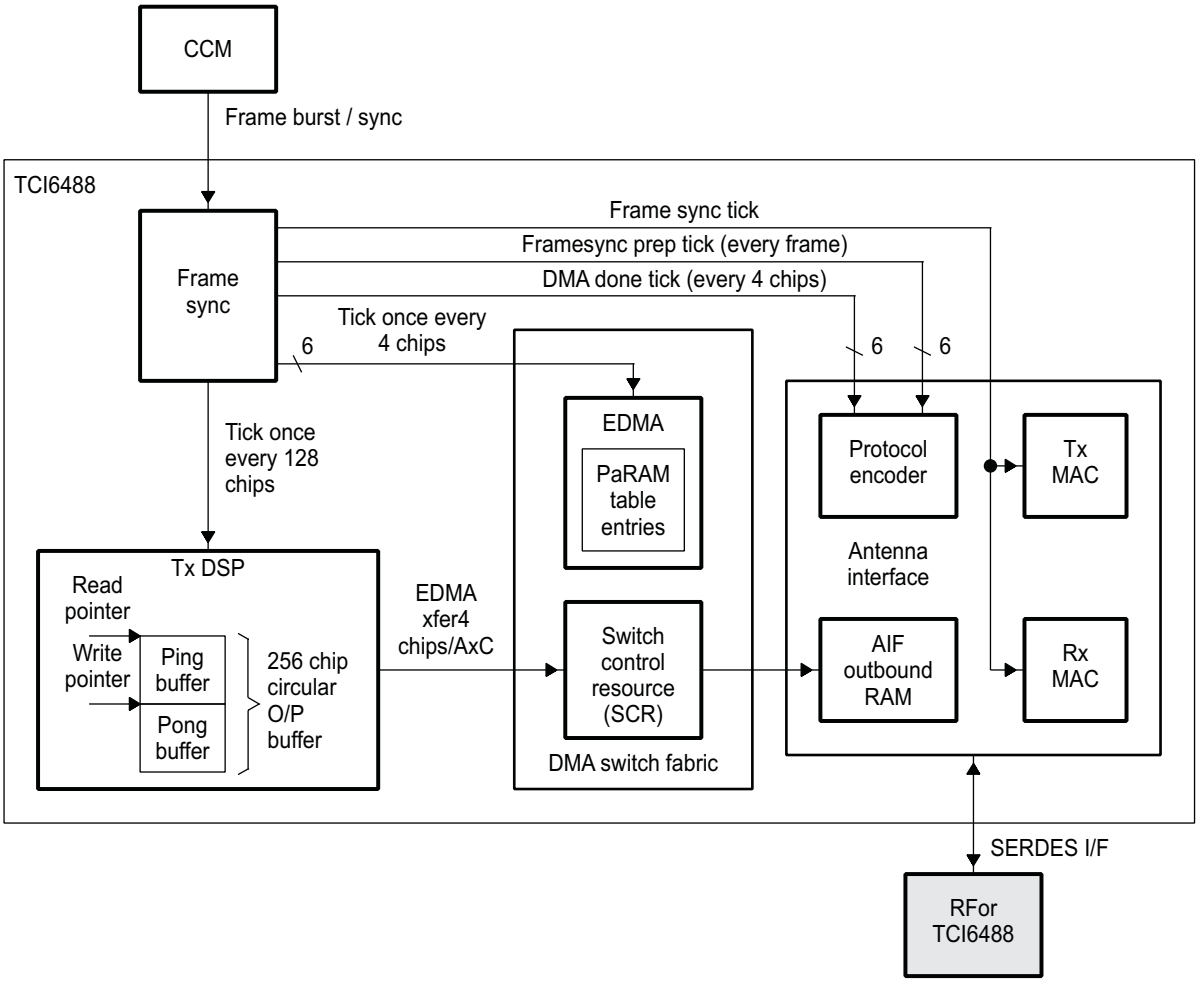
7.4 Setting Up Data Transfers

Setting up data transfers involves programming AIF configuration registers: frame sync and EDMA. There are no differences in programming frame sync and EDMA for OBSAI and CPRI. The differences are only involved in AIF setup. Circuit-switched data transfer configuration is the explained along with the differences with regards to the packet-switched transfers.

7.4.1 Software Configuration for DL Data Flow - DL DSP → AIF

Figure 61 shows the system-level overview of different modules involved in an outbound transfer from the Tx DSP to the AIF. The AIF interfaces with the RF card or another TMS320TCI6487/8.

Figure 61. Setup for Outbound Data Transfer from Tx DSP RAM to AIF Outbound RAM



The following sections discuss the configuration and run-time operation DL streams from the Tx DSP to the SERDES.

7.4.1.1 Configuration

The frame sync module receives its synchronization and clock signals from the system CCM module. These inputs are used to generate the frame sync tick to the antenna interface. The AIF Rx MAC and Tx MAC use the frame sync tick to synchronize the Rx/Tx SERDES links. Additionally, the frame sync is programmed to generate ticks to the AIF every four chips for a maximum of six links. The protocol encoder uses these four chip ticks to mark the completion of EDMA operation from Tx DSP to AIF. There is also a frame preparation tick for a maximum of six links. The protocol encoder uses these frame sync prep ticks to mark the beginning of a frame. The frame sync is also programmed to generate a tick that occurs every 128 chips for the TX DSP.

The Tx DSP allocates a circular buffer width of 256 chips (or two ping-pong buffers each with a width of 128 chips). In the figure, the read pointer initializes to the start of one of the ping/pong buffers. The write pointer offsets from the read pointer by 128 chips in the circular buffer.

The DMA switch fabric is configured with PaRAM entries so that data is transferred between the L2 RAM of the Tx DSP to the AIF RAM. The EDMA is configured to read four chips of data for all of the antenna streams when it receives a tick from the frame sync module. The tick from frame sync arrives every four chips. The DMA is set up using PaRAM entries and a combination of event chaining and link updates.

7.4.1.2 Operation

The Tx DSP receives a tick every 128 chips from the frame sync module. I/Q data is output in either the ping or pong buffers. The write pointer specifies which of the two buffers are used to write data. The write pointer is toggled to the start of the other buffer when data is finished writing into its current location (e.g., if the write location is the start of the pong buffer, then the write pointer points to the start of the ping buffer at the end of the Tx DSP operation.

The EDMA triggers to occur once every four chips. Four chips of outbound data are transfer for all antenna streams per link into the AIF outbound RAM. The EDMA must be programmed so that data is read from the correct output buffer (ping/pong). The EDMA starts transferring data from the Tx DSP by the location specified by the read location stored in the active EDMA PaRAM entry. The EDMA increments the read location by four chips.

The antenna interface runs a Tx frame synchronization protocol. Once the Tx link is in the FRAME_TX state, it grabs data from the AIF RAM and encodes/combines as necessary so that an OBSAI/CPRI complaint stream is produced at the SERDES output. Any synchronization required to put the Tx MAC in the FRAME_TX state must happen before the downlink is set up.

7.4.1.3 DL Set Up Steps

It is necessary to prepare the frame sync, Tx DSP, and AIF before you can set up a DL. The actions that you must perform to prepare the frame sync, Tx DSP, and AIF are listed in Section 7.4.1.3.1-Section 7.4.1.3.2. It is important that these actions are carried out prior to activating a DL in order to ensure that no loss of DL data occurs.

The order of the Tx DSP, frame sync, and AIF set up is not important.

It is unnecessary to set up frame sync for packet-switched transfers. AIF events trigger DL packet-switched transfers.

7.4.1.3.1 Tx DSP Set Up Actions

The points below describe the Tx DSP set up actions:

1. Set up ping-pong buffers in the Tx DSP memory for each link.
2. Configure the EDMA PaRAM entries to perform a data transfer from Tx DSP to AIF RAM.
3. The ping-pong buffer in the Tx DSP must be zeroed out to prevent any unwanted data from transmitting.
4. Set up and enable a trigger generator in the frame sync to tick the Tx DSP every 128 chips. This tick must be offset from the UMTS frame sync by the correct offset in order to compensate for delays in the processing chain. The Tx DSP uses this tick to spread/scramble 128 chips of data in the circular buffer. Initialization of the read and write locations in the circular buffer must occur in the Tx DSP init routine. The application software must also set up the ISR to service this tick.

7.4.1.3.1.1 Frame Sync Set Up Actions

Set up the RP3 and/or systems timer in the frame sync so that system events can generate. It is unnecessary to set up the frame sync event for packet-switched DL transfers because the AIF event can trigger the transfer (programming the frame sync is also an option to trigger the same).

7.4.1.3.2 AIF Set Up Actions

The points below explain the order in which the AIF is set up. The order of setup is important.

1. Zero the outbound AIF data buffer memory.
2. Set the frame sync up to generate a system event for AIF every ten msecs. The Tx MAC and Rx MAC modules use this tick in the AIF in order to synchronize with inbound and outbound links.

You must perform the steps below for every outbound link that must be set up:

1. Set up and enable the frame sync trigger generator to generate a frame sync preparation tick every ten msecs. The protocol encoder uses this tick to mark the beginning of a frame.
2. Set up and enable the frame sync trigger generators to trigger the EDMA every four chips. The EDMA transfers data from the Tx DSP memory to the AIF data buffer when it receives this tick. If two or more outbound links have the same timing, then you can use a single system event to perform the transfers.
3. Set up and enable the frame sync trigger generators to trigger the AIF for the DMA done event. The protocol encoder in the AIF uses this tick to pull data from the AIF data memory buffer.
4. The AIF PS event triggers the transfer of data from the inbound buffer for packet-switched data transfers. Hence, no Frame sync trigger is required.
5. Configure and enable outbound link(s).

The frame sync timer(s) are enabled when the above set up actions for AIF, Tx DSP, and frame sync are complete. The enabling of the timer(s) allows system-event generation and the Tx DSP operations to be performed.

7.4.1.4 Memory Organization

The memory organization for the Tx DSP L2 RAM and AIF RAM is shown in Figure 62. Certain assumptions have been made in Figure 62, primarily to demonstrate the EDMA parameters calculation.

7.4.1.4.1 DL Example Assumptions

Table 48. Assumptions Made to Demonstrate TMS320TCI6487/8 Set Up for DL

Description	Amount
Size of each Tx Buffer	128 chips
Number of Tx Buffers	2
Number of chips per EDMA transfer (Tx DSP → AIF)	4 chips
Number of bytes per chip	4

7.4.1.4.1.1 TX DSP L2 RAM

The TX DSP output buffer is organized as one continuous block of 128 chips for each antenna carrier. One output buffer is shown in Figure 62 for each antenna carrier for simplicity. There are two output buffers for each AxC to support read/write operations. Each chip on the output is a 16-bit I/Q chip, hence one chip occupies 32 bits (four bytes).

The memory allocation of the Tx DSP L2 RAM depends on the number of antenna carriers and on the size of each buffer (128 chips is chosen for this example).

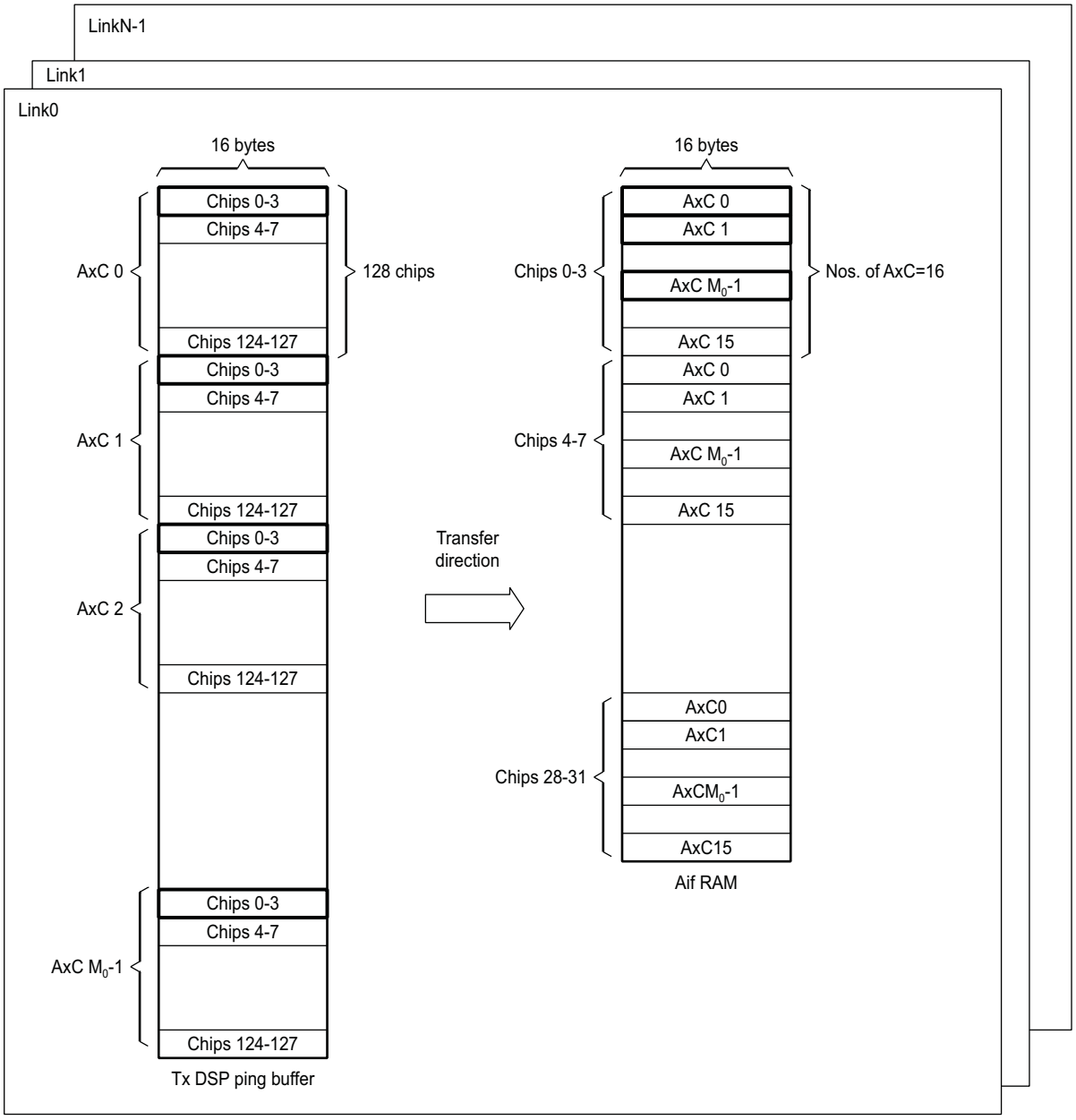
7.4.1.4.2 Antenna Interface Outbound RAM

The AIF RAM has fixed data storage space. It can store a maximum of 32 chips per antenna carrier and per link (each chip is four bytes on the outbound). Since 16 antenna streams must be supported for each link, there is a maximum of six outbound links. The total available antenna RAM to support outbound circuit-switched data is 16×6×32×4 = 12,288 bytes.

The AIF RAM is organized in blocks of four chips. Data for 16 antenna carriers are continuously placed in four-chip blocks.

The reason for this memory layout in the AIF is that it makes it convenient for the AIF to format these blocks to fit into the outbound SERDES link. The AIF memory allocation and organization do not change, even if the number of antenna carriers changes.

Figure 62. Memory Organization in Tx DSP Ping Buffer and AIF Outbound RAM

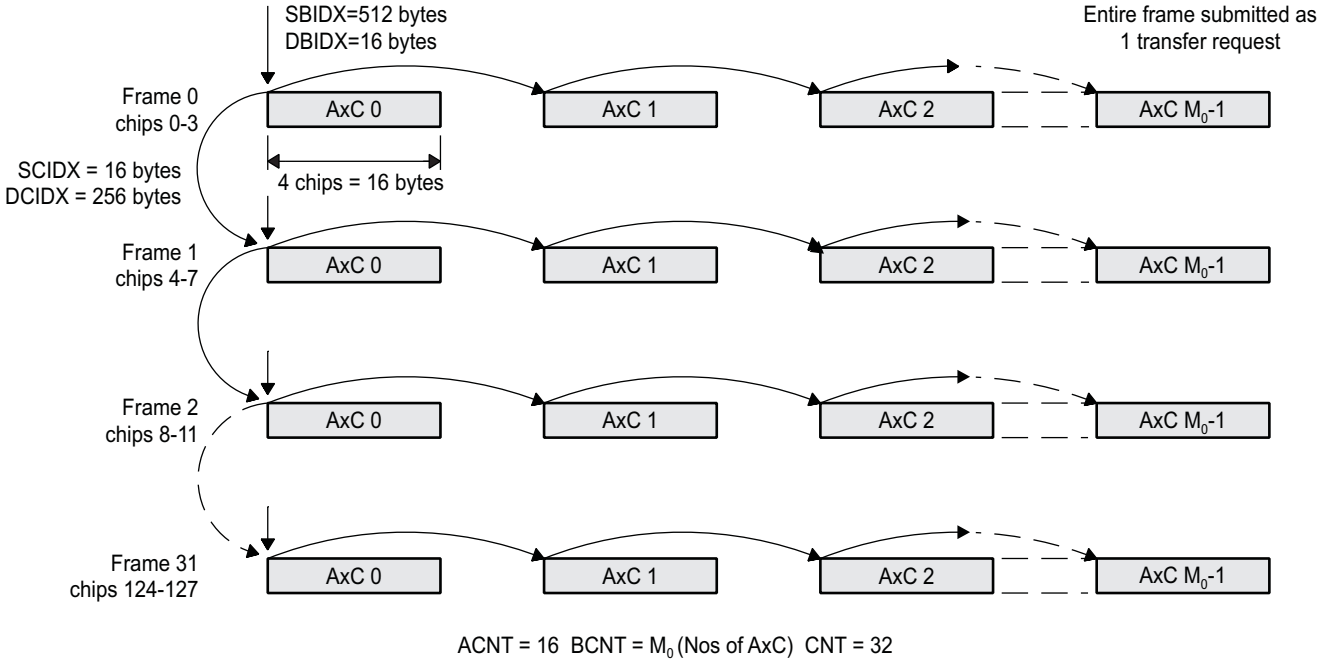


7.4.1.4.2.1 Transfer Parameters

Figure 63 shows the transfer for a single link. An AB-sync transfer is necessary to perform the transfer.

A system event (occurring every four chips) results in the transfer of a single frame for a link. A single frame contains four chips for all of the antenna carriers for a single link. Data that is transferred in the first EDMA transfer on receiving a system event from the frame sync is shown in dark lines in Figure 63.

Figure 63. Outbound DMA Transfer Shown for a Single Link



Each transfer request to the transfer controller transfers four chips for all of the antenna streams across all of the active links. At the of the last transfer request, 128 chips have been transferred for all antenna streams across all active links.

The important PaRAM entries are shown in Table 49.

Table 49. EDMA PaRAM Parameters for Outbound Transfer from Tx DSP to AIF Data Buffer

Parameter	Value	Description
ACNT	16	Number of bytes transferred in each array (= 4 chips according to OBSAI format)
BCNT	M	Number of antenna carriers in link
CNT	32	Number of frames (Size of output buffer / Size of array)
SBIDX	512	B dimension increment for source in bytes
DBIDX	16	B dimension increment for destination in bytes
SCIDX	16	C dimension increment for source in bytes
DCIDX	256	C dimension increment for destination in bytes
Intermediate chaining enabled	Yes	
Final chaining enabled	Yes	
Link update enabled	Yes	

The inbound packet-switch transfer parameters depend on the programmed depth of the PS FIFOs. Assume the FIFO depth is 2; then, ACNT =32, BCNT = 2, and CCNT = 960. The SBIDX and SCIDX values are zero since the DB circuitry keeps track of the read and write pointers in the packet-switched FIFOs and routes the actual VBUS reads to the appropriate address in the packet-switched FIFO.

The outbound packet-switched transfer parameters depend on the selected transmission rules. Assume that there is only one packet-switched slot in a message group, then the EDMA parameters are ACNT=32, BCNT=1, and CCNT=1920.

In order to perform a frame transfer for all links, the end of each frame transfer must be chained to the next link frame transfer. This is discussed further in the next section.

7.4.1.5 AIF Addressing Wrapping

The AIF can address 32 chips for a maximum of 96 antenna carriers. Since the L2 RAM is 128 chips for each antenna carrier, there would are four PaRAM entries for every 128 chips. If the L2 RAM is one frame (for delayed streams), the required number of PaRAM entries is much bigger than what is currently supported. Hence, the AIF wraps address accesses that map outside of the 32-chip memory space. Wrapping for up to 1.5 frames is supported.

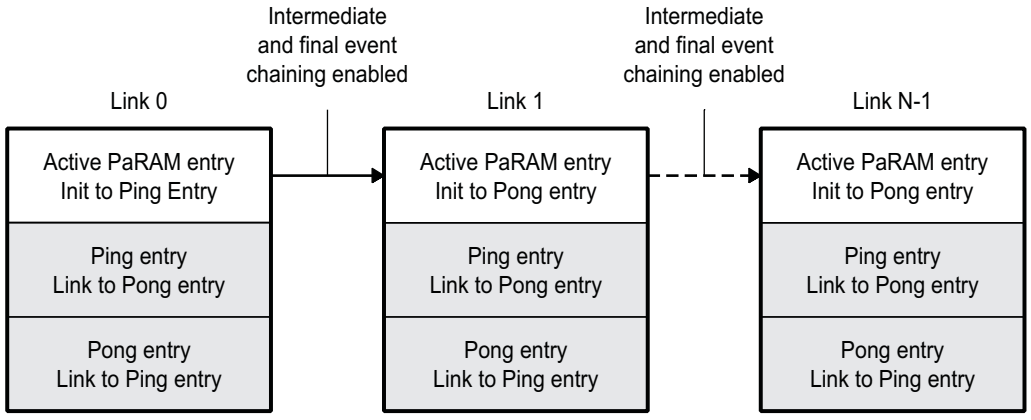
7.4.1.6 EDMA Chaining and Link Updates

A single EDMA PaRAM entry is required for each link. When a system event triggers the EDMA, it performs a transfer of four chips for all antenna carriers corresponding to the link's PaRAM entry; thus, a frame sync event generator must be configured for each link.

If two or more links have the same outbound timing, you can chain their PaRAM entries so that the same system event triggers the transfer of four chips for each link. In this case, intermediate and final event chaining must be enabled for the PaRAM entries, as shown in Figure 64.

Further, since Tx DSP uses a ping-pong buffer for each link to spread/scramble I/Q data, three PaRAM entries are required per link (an active entry, ping and pong entry). At initialization, the active PaRAM entry for each link is initialized to the PaRAM ping entry for that link. Following the final transfer request (end of 128 chips), the active PaRAM entry is updated to the pong PaRAM entry. Figure 64 shows how event chaining and link updates are used to transfer outbound data from Tx DSP L2 RAM to the AIF RAM.

Figure 64. Scheme showing event chaining and link updates used for transfer data from Tx DSP L2 RAM to AIF when outbound links have the same timing



7.4.1.7 EDMA PaRAM Entry Sizing

As discussed in Section 7.4.1.6, three PaRAM entries are required per link in order to support the ping-pong buffer operation. This remains the same for both packet-switched and circuit-switched transfers.

Table 50. EDMA PaRAM Entry Sizing for Transfer from Tx DSP to Outbound AIF RAM

Number of PaRAM channels per outbound link	1

7.4.1.8 EDMA Event Sizing

Each link must have its own DMA channel (and event) associated with the EDMA of outbound data. This remains the same for both packet-switched and circuit-switched transfers.

Table 51. EDMA Event Sizing for Transfer from Tx DSP to Outbound AIF RAM

Number of EDMA channels per outbound link	1

7.4.1.9 Frame Sync Event Sizing

You must program the frame sync trigger generators to cause events that facilitate the data transfer from Tx DSP memory to AIF. Table 52 shows the number of trigger generators that are necessary to support the data transfer from the Tx DSP to the AIF data buffer.

Table 52. Frame Sync Event Sizing for Transfer From Tx DSP to AIF Outbound RAM

Description	Number
Events to Tx DSP	1
Event to AIF (frame sync tick)	1
Events to EDMA per link	1
Events to AIF per link (circuit switched transfer)	2
Maximum number of event generator circuits required per outbound link	5

7.4.1.10 Tx Buffer Sizing Considerations

The following considerations must be taken into account when the size of the Tx buffer is determined:

- A larger buffer improves Tx DSP performance since it interrupts the CPU less often (e.g., if the Tx DSP buffer in Figure 62 is increased from 128 to 512 chips, it is required to interrupt the Tx DSP every 512 chips); however, it increases the latency for closed-loop signaling.
- A smaller buffer decreases the closed-loop latency, but the Tx DSP performance can be affected since the DSP is interrupted more frequently and a larger portion of time is dedicated to interrupt servicing.
- The buffer must be sized so that all DL processing actions complete before the next DL processing start tick arrives.

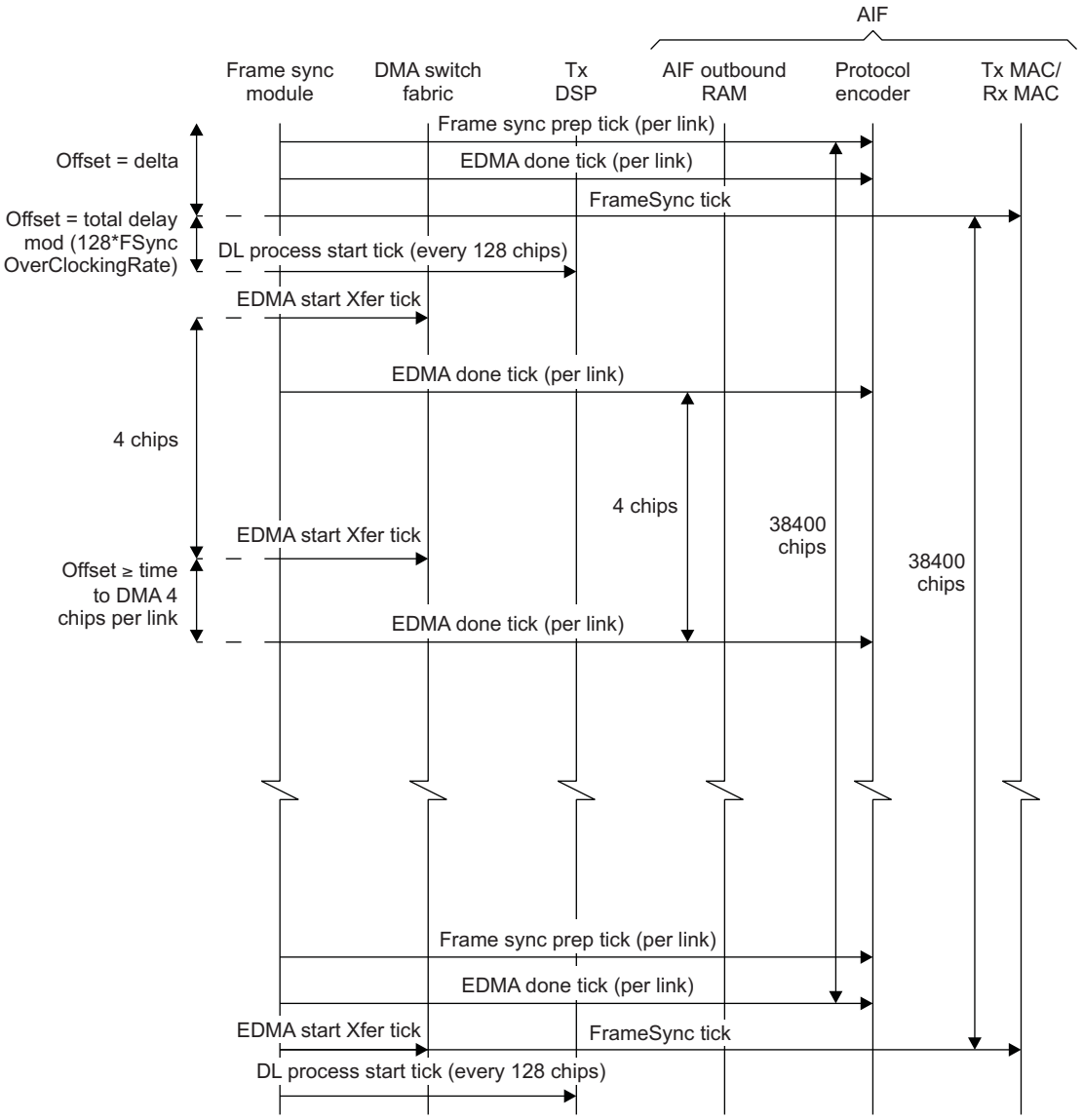
7.4.1.11 DL Event Timing Relationships

Figure 65 shows the timing relationships for different system events that is required to set up a DL operation. You should choose the offsets so that the closed-loop latency in the system is minimized. The following points should be considered:

- The AIF receives the EDMA done tick for an active link before the EDMA start transfer tick for the next four chips transfer. To achieve this, program the trigger generators so that the offset between the start and done tick is greater than or equal to the time that it takes to perform the EDMA transfer from Tx DSP to AIF for a single link.

- The offset between the frame tick and the DL process start tick is determined by the total delay in the system. The total delay is the combination of FIFO delays in the AIF data processing chain, EDMA delays and delays caused due to the chaining of TMS320TCI6487/8 units. The offset between the frame tick and the DL process start tick is modulo128×frame sync over-clocking rate, where 128 is the periodicity of the DL ticks to the Tx DSP and the frame sync over-clocking rate (this can be 1, 2, 4, or 8).
- The DL process start event can occur after the EDMA start event since downlink processing and the EDMA transfer operate on different buffers (ping or pong).
- The offset between the frame sync prep event and the frame sync tick is specified by Δ . The Δ offset is equal to the latency in the TMS320TCI6487/8 daisy of the outbound link from the current TMS320TCI6487/8 to the end point.

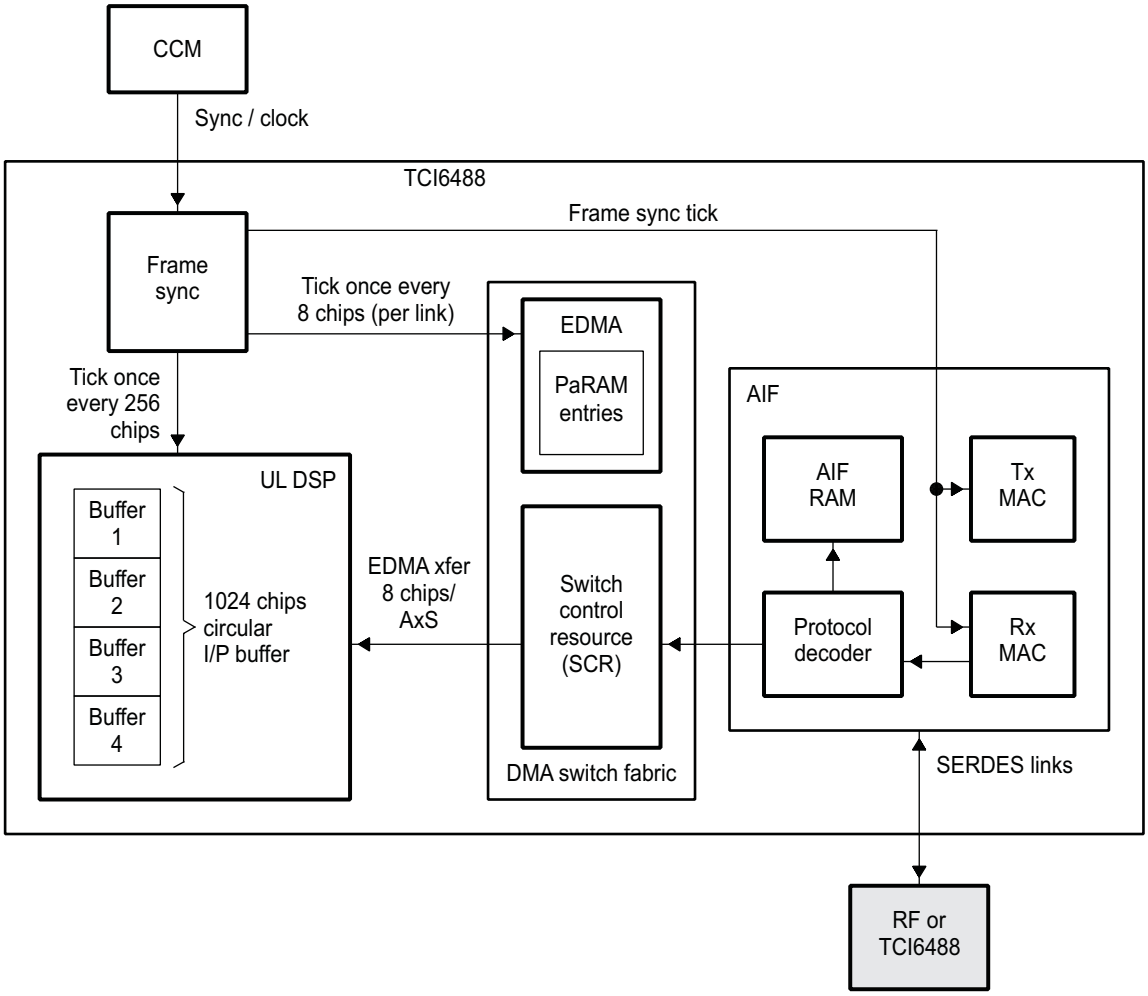
Figure 65. Event Timing Relationships Shown for Outbound Transfer from Tx DSP to AIF Outbound RAM for a Single Link



7.4.2 Software Configuration for UL Data Flow - AIF → UL DSP

Figure 66 shows the system-level set up for different modules involved in an inbound transfer from AIF to UL DSP. The AIF may interface to another TMS320TCI6487/8 or to the RF card.

Figure 66. Inbound Transfer From AIF Inbound RAM to UL DSP L2 RAM



7.4.2.1 Configuration

The frame sync module receives its clock and synchronization signals from the CCM. The frame sync events generator is set up to provide the AIF with a frame sync tick. The Rx MAC and the Tx MAC use the frame sync tick to synchronize the Rx/Tx SERDES links. In addition, the frame sync is programmed to provide a tick to the EDMA every eight chips. This eight-chip tick causes the EDMA to transfer eight chips of data from the AIF RAM to the UL DSP L2 buffer. The eight-chip tick to the EDMA must be set up for every link. The frame sync event generators must also provide a tick to the UL DSP.

In the set up shown in Figure 66, the UL DSP receives a tick once every 256 chips. The UL DSP processes 256 chips of data every time it receives this tick. The UL DSP can perform one or more of the following tasks for each tick: path detect, path monitor, or finger de-spread operations, depending on the set up of the system. The UL DSP must buffer radio data in order to perform its processing (path detect, path monitor, or finger de-spread). The amount of buffering is discussed in later sections. In the example above, four buffers are configured per link, each containing 256 chips. The EDMA uses one of the four buffers to transfer data from the AIF RAM, hence the UL DSP can only use three buffers.

7.4.2.2 Operation

The UL DSP receives a tick every 256 chips from the frame sync module. The UL DSP processes 256 chips of data every tick and it can perform one or more of the following operations: path detect, path monitor, or finger de-spread. The amount of buffering that is required is a design consideration. At least four buffers are required: one buffer to perform EDMA of radio data from the AIF and three buffers are required to perform any of the UL processing tasks: path detect, path monitor, or finger de-spread.

The frame sync provides an eight-chip tick per link to the EDMA. Upon receiving this tick, the EDMA transfers eight chips worth of data for all active antenna streams for the link from AIF RAM to L2 RAM in the UL DSP. The PaRAM entries ensure that the antenna streams are being transferred in the correct L2 buffers. The frame sync event generators must be programmed with the correct offset to ensure that the latency between the chips arriving in the AIF and being transferred to the L2 RAM.

7.4.2.3 UL Set Up Steps

The subsections below explain the steps necessary to set up the UL chain. The set up actions involve setting the UL DSP, frame sync, and the AIF. It is important that these actions are carried out before any UL processing is performed to prevent any loss of data. Any of the three GEM cores on the TMS320TCI6487/8 can perform the frame sync and AIF set up actions. The order of the UL DSP, frame sync, and AIF set up is not important.

7.4.2.3.1 UL DSP Set Up Actions

The set up steps below explain the UL DSP set up actions. The order of these actions is important.

1. Set up the radio data buffers. A minimum of four buffers is required: one buffer is required to DMA data from AIF to UL DSP and the other three buffers are used to perform UL processing tasks.
2. Zero-out the contents of the radio buffers.
3. Set up the EDMA PaRAM entries to perform the data transfer from AIF to UL DSP.
4. Set up and enable an frame sync event generator so that the UL DSP receives a system event every 256 chips.

7.4.2.3.2 Frame Sync Set Up Actions

Set up the RP3 and/or system timer in frame sync so that system events can generate.

7.4.2.3.2.1 AIF Set Up Actions

1. Zero the inbound AIF data memory.
2. Set up and enable the frame sync trigger generator to generate a system event for AIF every 10 msecs. The Tx MAC and Rx MAC in AIF use this tick for synchronizing with inbound and outbound links. You can skip this step if it is already set up.
3. for every link that is set up, you must set up and enable the frame sync trigger generator to trigger the EDMA to perform a transfer of eight chips of data from AIF RAM to UL DSP memory.

The frame sync timer(s) are enabled when the above set up actions for AIF, UL DSP, and frame sync are complete. Enabling the timer(s) allows system event generation and the UL DSP operations occur.

7.4.2.4 Memory Organization

The memory organization for the AIF RAM and UL DSP L2 RAM are shown in Figure 67.

7.4.2.4.1 UL Example Assumptions

In order to demonstrate the memory organization for UL data transfer, the assumptions in Table 53 are made.

Table 53. Assumptions Made to Demonstrate TMS320TCI6487/8 Set Up for UL

Description	Amount
Size of each UL Buffer	256 chips
Number of UL buffers per over-sampled antenna stream	4
Number of chips per EDMA transfer (AIF → UL DSP)	8 chips
Number of bytes per chip	2
AxS over-sampling factor	2
Number of antenna streams for link 0	M_0
Number of buffers per link	M0 × Number of UL Buffers × AxS over-sampling Factor

It is assumed that all antenna streams for all active links are on the same DSP. You can have different streams stored on different GEM cores. However, it is recommended that you store antenna streams on one link on the same DSP to reduce the PaRAM entries.

7.4.2.4.2 UL DSP RAM

The memory organization for UL DSP RAM is shown in Figure 67. Each over-sampled antenna stream needs at least four buffers to store I/Q data. These four buffers are arranged continuously in memory. One buffer is used to DMA data from the AIF RAM and the UL DSP can use the remaining three buffers for performing path detect, path monitor, or finger de-spread. Each buffer stores 256 chips of an over-sampled antenna stream. Each I/Q sample is eight bits (each chip is two bytes). The total number of buffers that are required for a link that has M0 antenna carriers is M0×(number of Rx buffers per over-sampled antenna stream)×(AxS over-sampling factor).

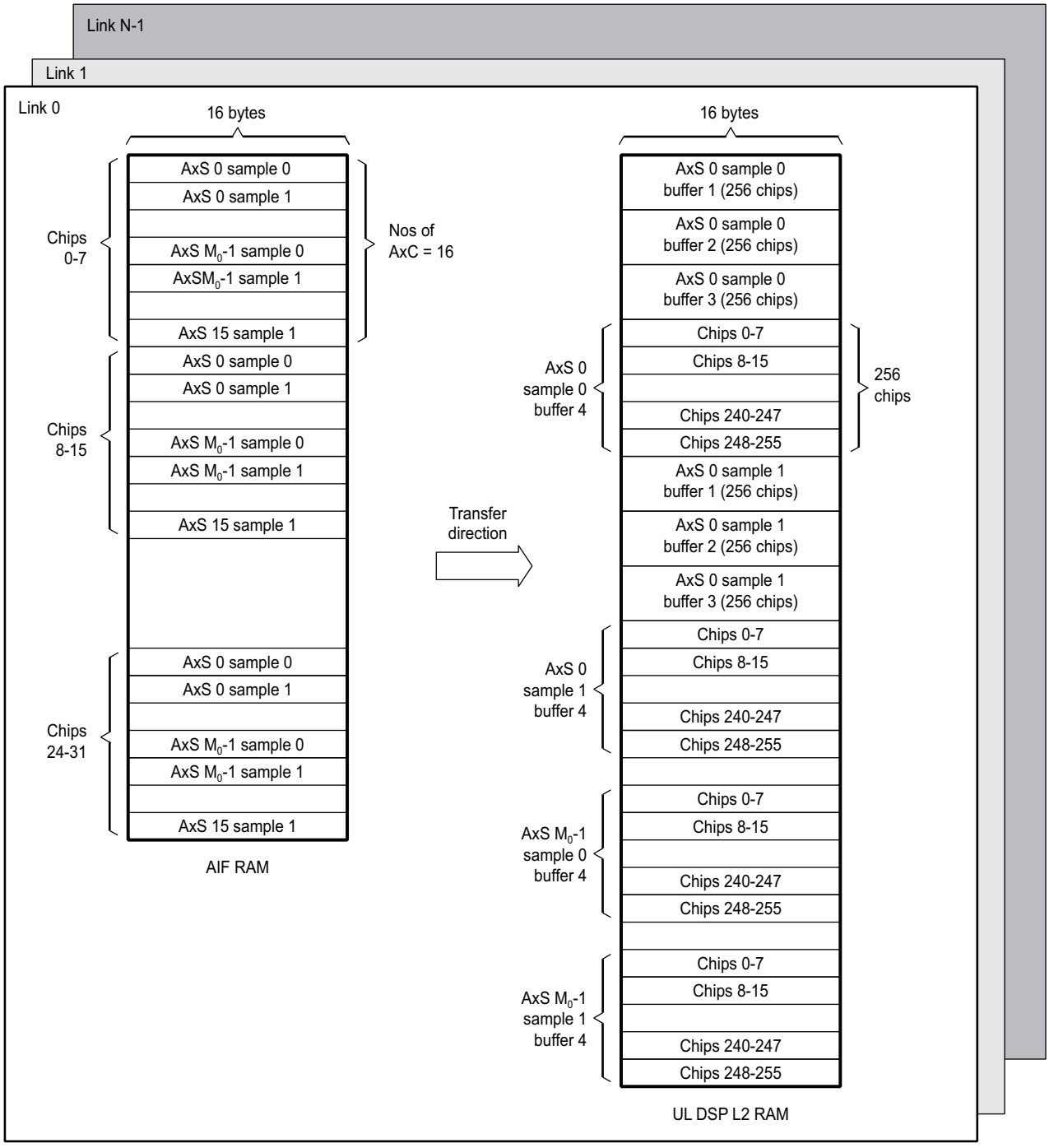
The amount of memory required in the UL DSP to store antenna streams depends on the parameters listed in Table 53.

Packet-switched data is transferred from the DSP to a packet-switched outbound buffer.

7.4.2.4.3 Antenna Interface Inbound RAM

The memory arrangement for storing inbound antenna streams in the AIF RAM is shown in Figure 67. The data format shown is the UL RSA format. This format is optimized for RSA processing. The AIF inbound RAM can store a maximum of 32 chips (each chip is two bytes) for each over-sampled antenna stream. Currently, only antenna streams may be over-sampled by a factor of two. Since there are 16 antenna streams per link, and there are a maximum of six links, the total available AIF inbound RAM is 16×6×32×2×2 = 12,288 bytes. Therefore, the amount of AIF inbound is equal to that available for outbound links.

Figure 67. Memory Organization in AIF Inbound RAM and UL DSP L2 RAM



7.4.2.5 Transfer Parameters

Figure 68 shows the inbound transfer of UL data for a single link. An AB sync transfer performs the transfer. The figures for the EDMA transfer are based on the assumptions listed in Table 53.

The EDMA is triggered every eight chips to perform the data transfer from the AIF RAM to one of the four buffers in the UL DSP. The EDMA is programmed to transfer eight chips of data for each over-sampled antenna stream for all active antenna streams across all active links. Data that is transferred in the first EDMA transfer on receiving a system event from the frame sync is shown in dark lines in Figure 68.

Figure 68. Inbound DMA Shown for AIF → UL DSP Transfer for a Single Link

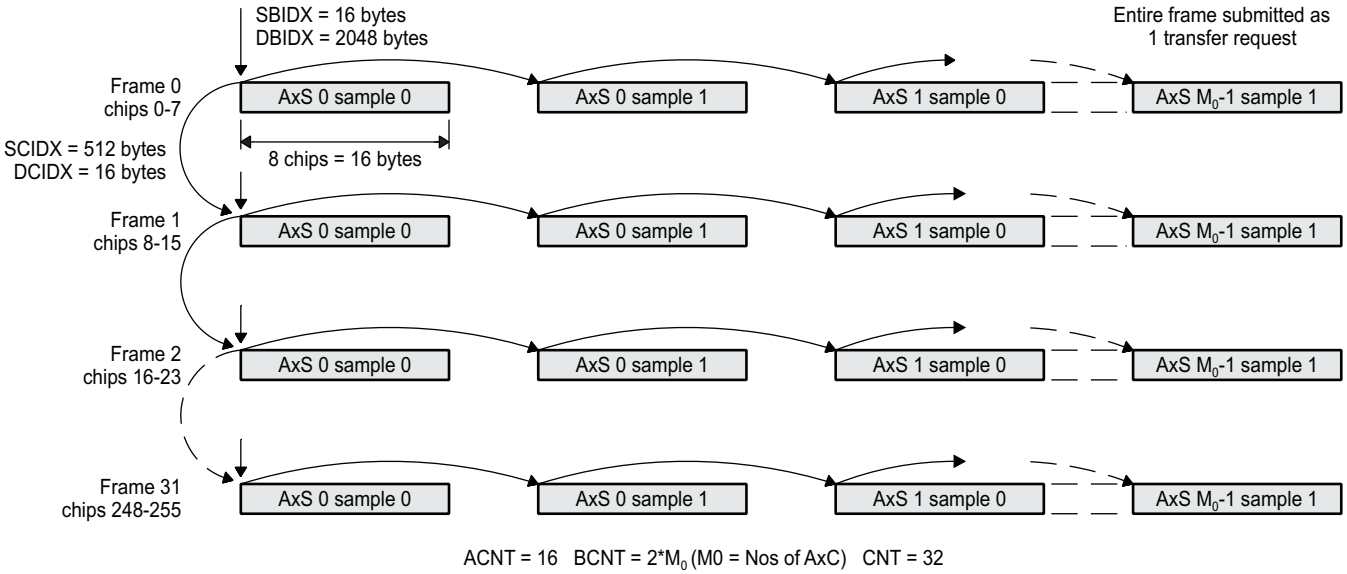


Table 54. EDMA PaRAM Entries for Inbound Transfer From AIF RAM to UL DSP

Parameter	Value	Description
ACNT	16	Number of bytes transferred in each array (each array consists of 8 chips, and 1 chip occupies 2 bytes on UL)
BCNT	$2*M_0$	Over-sampling factor on UL × Num of AxS per link
CNT	128	Number of TR frames (Size of buffer in chips × Nos. of buffer per AxS ÷ Size of array in chips)
SBIDX	16	B dimension increment for source in bytes
DBIDX	2048	B dimension increment for destination in bytes
SCIDX	512	C dimension increment for source in bytes
DCIDX	16	C dimension increment for destination in bytes
Intermediate chaining enabled		See Section 7.4.1.6.
Final chaining enabled		See Section 7.4.1.6
Link update enabled	Yes	See Section 7.4.1.6

The Packet-switched data parameters are:

- ACNT = 32 (19 bytes of control data and remaining 13 bytes zeros, since the packet switch fifo is 16 byte wide, so we need to transfer two 16 bytes of data)
- BCNT = 1
- CCNT = 8

In order to perform a frame transfer for all links, the end of each frame transfer must be chained to the next link frame transfer. This is discussed in Section 7.4.2.7.

7.4.2.6 AIF Address Wrapping

The AIF inbound RAM can store 32 chips per over-sampled antenna stream and the buffer size of UL DSP can be larger than 32 chips. In this example, the buffer size in the UL DSP is 256 chips. One PaRAM entry is required per 32 chips of transfer. This requires a large number of PaRAM entries for the EDMA.

In order to reduce the number of PaRAM entries, the AIF supports address wrapping for up to 1.5 frames. The address wrapping feature means that one EDMA PaRAM entry is required per link. The buffer size can be as large as 1.5 frames. In the example set up shown in Figure 69, a single PaRAM entry is used to DMA 1,024 chips (= number of buffer per AxS × number of chips per buffer).

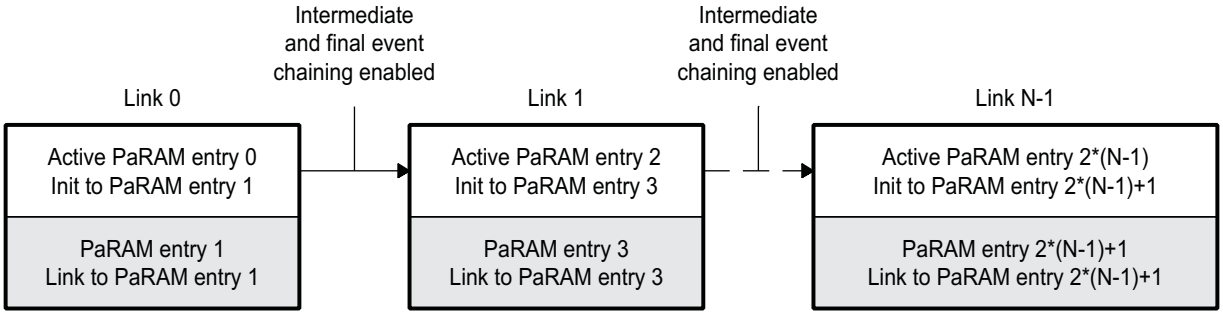
7.4.2.7 EDMA Chaining and Link Updates

A single EDMA PaRAM entry is required for each link. An frame sync event generator must be configured for each link so that when a system event triggers the EDMA, it can perform the transfer of eight chips for all antenna streams that correspond to the link's PaRAM entry.

If two or more links have the same inbound timing, then you can chain their PaRAM so that the same system event triggers the transfer of eight chips for each link. In this case, intermediate and final event chaining must be enabled for the PaRAM entries, as shown in Figure 69.

When each PaRAM entry is exhausted, it must be re-loaded so that the DMA can continue indefinitely when system events trigger the EDMA. Hence, in addition to one active PaRAM entry per link, a re-load entry must be supplied. So, two PaRAM entries are required per link for the example shown. In order to support the re-loading feature, the link update feature in the PaRAM entry must be enabled.

Figure 69. Scheme Showing Event Chaining and Link Updates for Transfer Data From AIF Inbound RAM to UL DSP RAM for Inbound Links That Have the Same Inbound Timing



7.4.2.8 EDMA PaRAM Entry Sizing

Two EDMA PaRAM entries are required per inbound link.

Table 55. EDMA PaRAM Entry Sizing for Transfer From AIF Inbound RAM to UL DSP RAM

Number of PaRAM entries per inbound link	2

7.4.2.9 EDMA Event Sizing

Each link must have its own DMA channel (and event) associated with the EDMA of inbound data.

Table 56. EDMA Event Sizing for Transfer From AIF Inbound RAM to UL DSP RAM

Number of EDMA channels per inbound link	1

7.4.2.10 Frame Sync Event Sizing

The frame sync event generators must be programmed to cause events that facilitate the transfer of data from AIF RAM to the UL DSP. Table 57 shows the number of trigger generators that are required to support the data transfer from AIF RAM to UL DSP RAM.

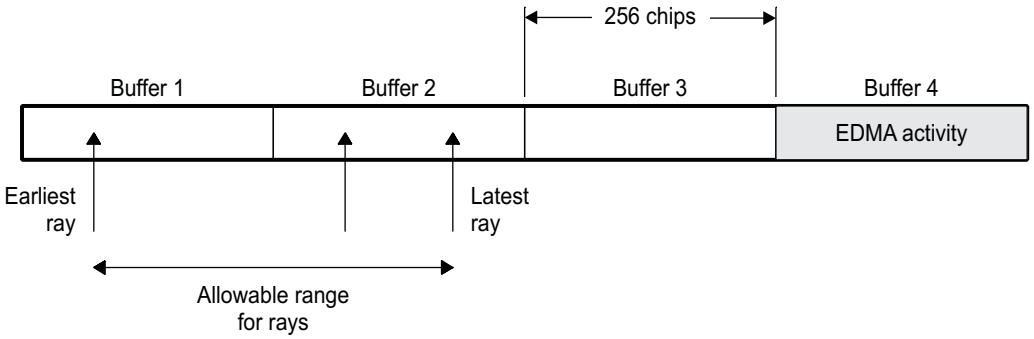
Table 57. Frame Sync Event Sizing for Transfer From AIF Inbound RAM to UL DSP RAM

Description	Amount
Events to UL DSP	1
Event to AIF (frame sync tick)	1
Events to EDMA per link	1
Maximum number of event generator circuits required per outbound link	3

7.4.2.11 Rx Buffer Sizing Considerations

Figure 70 shows how the rays are managed within the four Rx buffers. A ray marks the start of the UL frame for a particular user. Multiple rays can be associated with a single user due. Any finger allocation algorithm must allocate rays within the allowable range for rays, as shown in Figure 70. This design allows for handling rays from different users and allowing for some ray drift. Also, three of the four buffers are used for path detect, path monitor, or finger de-spread operations and the fourth buffer is used for EDMA activity.

Figure 70. Rx Buffer Sizing Considerations



In Figure 70, the size of each Rx buffer is 256 chips. The Rx buffer contains I/Q data for a single over-sampled antenna stream. The design considerations for determining Rx buffer size in the UL DSP is similar to that of the DL buffer size in the Tx DSP (see Section 7.4.1.10).

The main points are summarized below:

- A larger buffer improves UL DSP performance for path detect, path monitor, or finger de-spread operations since it will be interrupted less often; however, it increases the latency for closed-loop signaling.
- A smaller buffer decreases the closed-loop latency, but a smaller buffer can affect the UL DSP since the DSP is interrupted more frequently and a larger portion of time is dedicated to interrupt servicing.
- The buffer must be sized so that all UL processing actions are completed before the next UL processing start tick arrives.
- The combined size of the buffers should be reflective of the maximum allowable range for rays. For example, if the allowable range for rays is 300 chips, then the size of all of the buffers cannot be less than 300 chips.

7.4.2.12 UL Event Timing Relationships

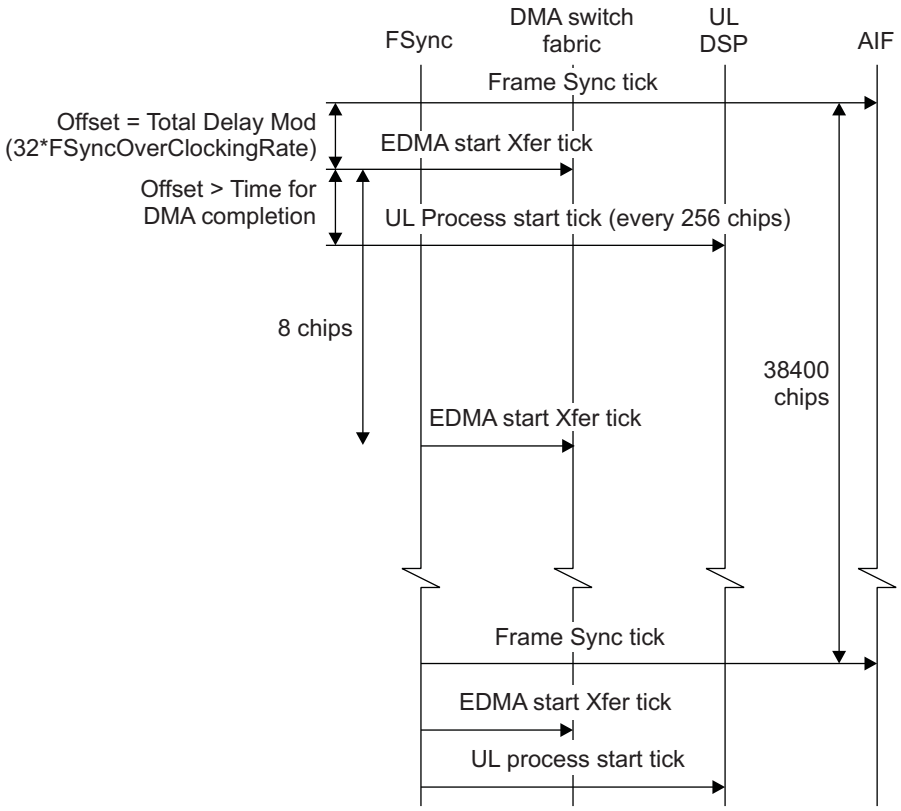
Figure 71 shows the timing relationships for different system events that are required to set up a UL operation. Choose the offsets so that the closed-loop latency in the system is minimized.

Consider these points:

- The EDMA start tick is programmed in a way that minimizes the closed-loop latency in the system (the latency in transferring data from inbound RAM in AIF to the DSP is minimized). To reduce this latency to a minimum, program the event generator offset in the frame sync to be (total delay) mod (32×frame sync over-clocking rate) where 32 chips is the size of the inbound AIF RAM and frame sync over-clocking rate is the input clock rate to the frame sync divided by the chip rate (frame sync over-clocking rate = frame sync clock rate ÷ 3.84 MChips/sec). The total delay covers the time it takes from origin of the data sample to the time it takes to write to the AIF inbound RAM.
- If you configure the TMS320TCI6487/8in a daisy chain, the total delay is inclusive of all of the latencies in the chain (the total delay is the time takes for a data sample go from its origin to the RAM of the TMS320TCI6487/8 under consideration). The TMS320TCI6487/8 data sheet provides a value for the latency that covers the time it takes for a data sample to arrive at the Rx MAC and be written to the inbound AIF RAM.

- The frame sync event generator must be programmed so that the UL process start tick is programmed with an offset that is greater than the time it takes to DMA eight chips of data from the AIF inbound RAM to the DSP L2 memory. This is necessary because at the end of every 256 chips, the buffer is used for UL processing (see Figure 71). Any race condition between using data still being DMA'ed and the buffer being used by the UL process can be avoided if this condition is satisfied.

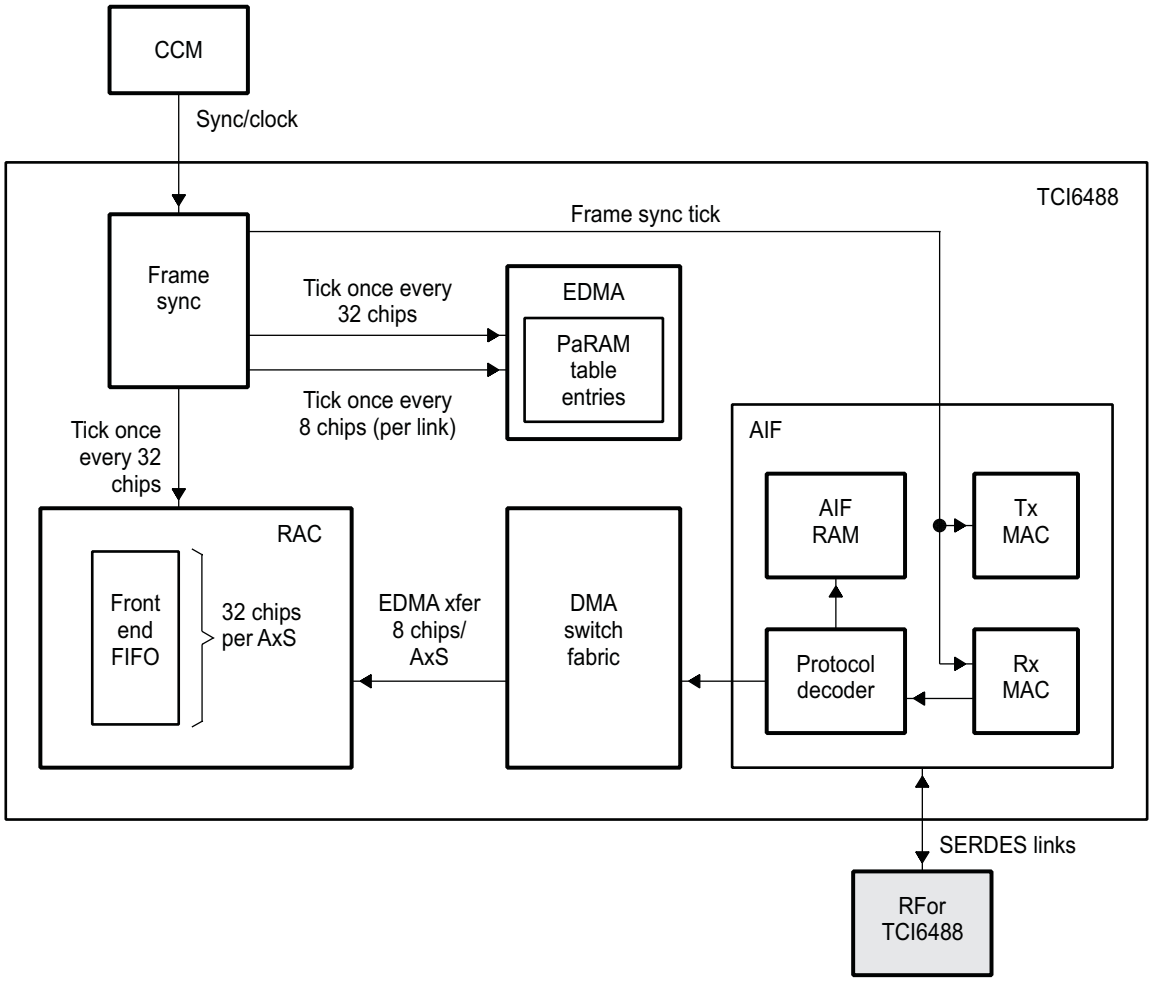
Figure 71. Event Timing Relationships Shown for Inbound Transfer from Inbound AIF RAM to UL DSP L2 RAM for a Single Link



7.4.3 Software Configuration for UL Data Flow - AIF → RAC

The system-level set up involving different modules performing an inbound transfer from AIF to RAC is shown in Figure 72.

Figure 72. Set Up for Inbound Transfer from AIF to RAC



7.4.3.1 Configuration

The frame sync module receives its clock and synchronization signals from the CCM. The frame sync event generator is set up to provide the AIF with the frame sync tick occurring every 38,400 chips (10msecs). Additionally, the frame sync is programmed to generate two system events to the EDMA: one of these events occurs every 32 chips and the second system event occurs every eight chips. The frame sync is also programmed to generate a system event to the RAC occurring every 32 chips.

7.4.3.2 Operation

The inbound data flow from AIF to RAC operation requires these steps:

1. The EDMA transfers a time stamp from the frame sync to the RAC every 32 chips.
2. The time stamp transfer is followed by four EDMA transfers of data from AIF inbound RAM to the RAC front-end FIFO. Each transfer occurs every eight chips and transfers eight chips of I/Q data for up to 54 over-sampled antenna streams (over-sampling factor = 2). This is intended to allow 48 on-time streams and six delayed streams.
3. The frame sync provides a system event to start the RAC operation when the EDMA transfer is complete. This system event occurs every 32 chips and must arrive at the RAC when the EDMA transfers for the 32 chips are complete.

7.4.3.3 RAC Set Up Steps

The subsections below explain the steps to set up an inbound transfer from AIF to RAC. The set up actions involve setting up the frame sync, AIF, and RAC; the order of the set up is irrelevant.

Perform the set up actions prior to performing any UL processing in order to prevent any loss of data. Any one of the three GEM cores on the TMS320TCI6487/8 can perform the frame sync and the AIF set up actions.

7.4.3.3.1 RAC Set Up Actions

The following steps are used to set up RAC:

1. Zero out the contents of the RAC input FIFO.
2. Set up the EDMA PaRAM entry to perform timestamp transfer from the frame sync memory-mapped register to the RAC register.
3. Configure and enable the frame sync event generator to generate a system event to the EDMA occurring every 32 chips for the time stamp transfer.
4. Configure and enable an frame sync event generator to provide a start signal to the RAC every 32 chips.

7.4.3.3.1.1 Frame Sync Set Up Actions

Set up the RP3 and/or system timer in frame sync so that system events can generate.

7.4.3.3.1.1.1 AIF Set Up Actions

The following steps are used to set up AIF:

1. Zero out the inbound memory
2. Set up the frame sync tick so that a system event generates every 10msecs. The Tx MAC / Rx MAC in AIF uses this tick to synchronize with external links. You can skip this step if it has already been set up.
3. Perform the following steps for every link that is set up:
   - Configure a PaRAM entry so that the EDMA can perform data transfers from AIF to RAC.
   - Configure and enable an frame sync event generator to cause a system event to the EDMA that occurs every eight chips to perform data transfer from AIF inbound RAM to the RAC.
4. In Step 1 above, the EDMA set up must precede the frame sync event generator set up if the links are added dynamically.
5. The frame sync timers (RP3 and/or system timer) are enabled when the set up steps outlined above are completed. The enabling of timer(s) allows for system event generation and the performance of periodic actions listed in Section 7.4.4.4.

7.4.3.4 Memory Organization

7.4.3.4.1 RAC Example Assumptions

In order to demonstrate memory organization for RAC data transfer, a few assumptions are required.

They are listed in Table 58.

Table 58. Assumptions Made to Demonstrate TMS320TCI6487/8 Setup for RAC

Description	Amount
Number of chips per EDMA transfer (AIF → RAC)	8 chips
Number of bytes per chip	2
AxS over-sampling factor	2
Number of Antenna Streams for Link 0	M_0

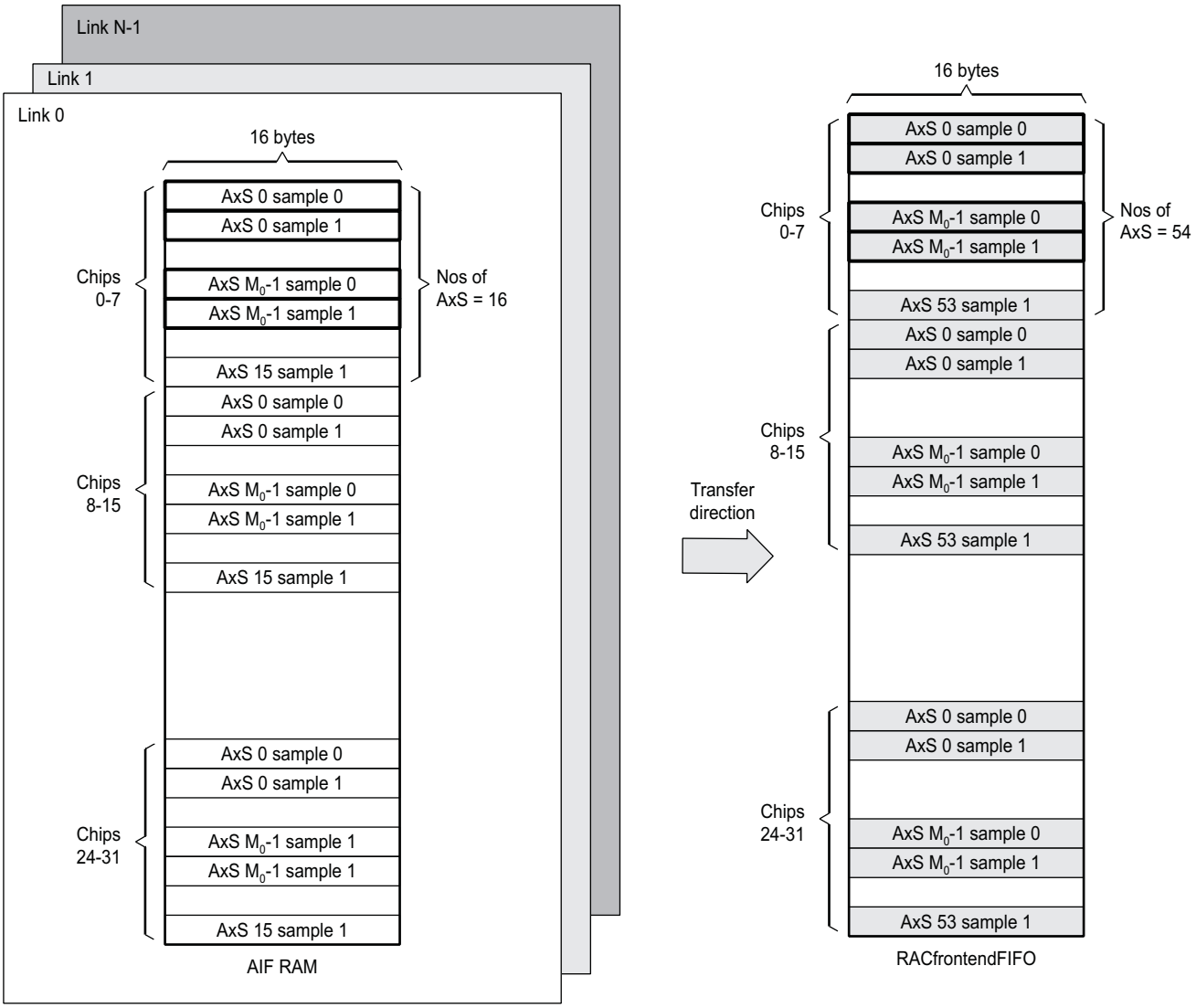
7.4.3.4.2 AIF Inbound RAM

The memory arrangement for storing inbound antenna streams in the AIF RAM is shown in Figure 73. The AIF inbound RAM can store a maximum of 32 chips (each chip is two bytes) for each over-sampled antenna stream. Currently, only antenna streams can be over-sampled by a factor of two. Since there are 16 antenna streams per link, and there are a maximum of six links, the total available AIF inbound RAM is 16×6×32×2×2 = 12,288 bytes. Therefore, the amount of AIF inbound is equal to that available for outbound links (see Section 7.4.1.4.2). The UL RSA format is required for RAC use.

7.4.3.4.3 RAC Front End FIFO

The RAC front end is a FIFO that can hold up to 54 antenna streams. Figure 73 shows the memory arrangement in the RAC FIFO. Each over-sampled antenna stream is arranged in eight-chip blocks of 16 bytes (one chip = two bytes). Each EDMA transfer requires a transfer of eight chips for each over-sampled AxS every eight chips. The FIFO can hold 32 chips for up to 54 over-sampled antenna streams. The total available memory in RAC FIFO is 54×32×2×2 = 6,912 bytes.

Figure 73. Memory Organization in RAC FE FIFO and AIF Inbound RAM



7.4.3.5 Transfer Parameters

Figure 74 shows the inbound transfer of UL data for a single link. An AB sync transfer performs the transfer. The figures for EDMA transfer are based on assumptions listed in Table 58.

The EDMA is triggered every eight chips to perform the data transfer from the AIF RAM to the RAC FE FIFO. The EDMA is programmed to transfer eight chips of data for each over-sampled antenna stream for all active antenna streams across all active links. Data that is transferred in the first EDMA transfer on receiving a system event from the frame sync is shown in dark lines in Figure 74.

Figure 74. DMA Shown for Inbound AIF RAM → RAC Transfer for a Single Link

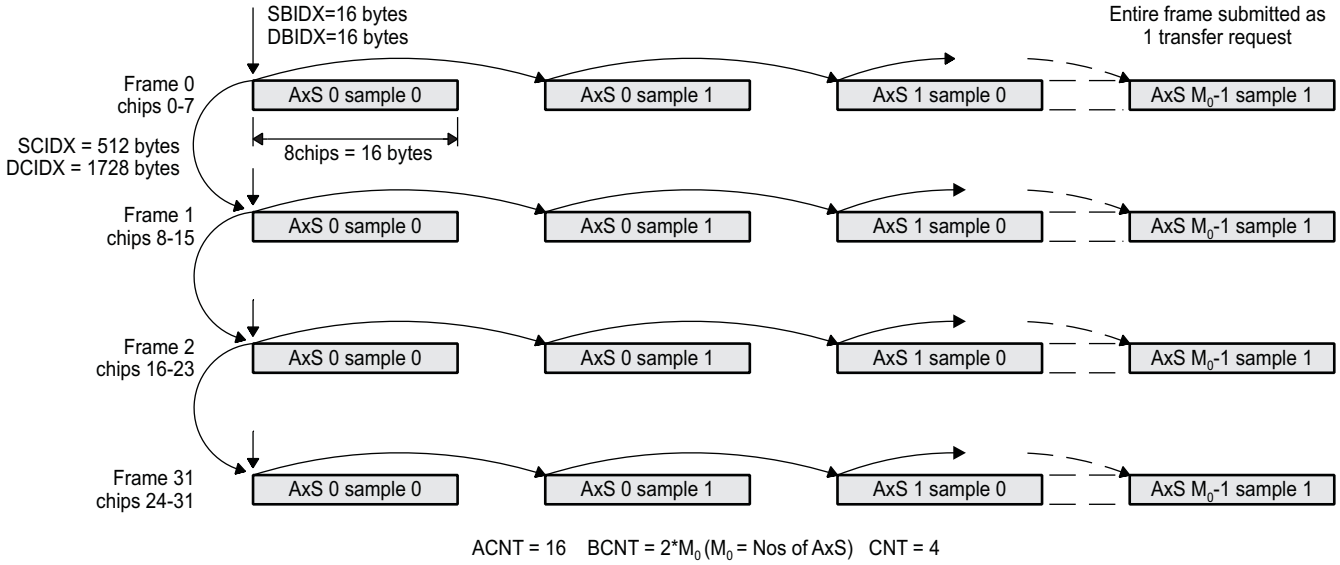


Table 59. EDMA PaRAM Entries for Transfer From Inbound AIF RAM to RAC

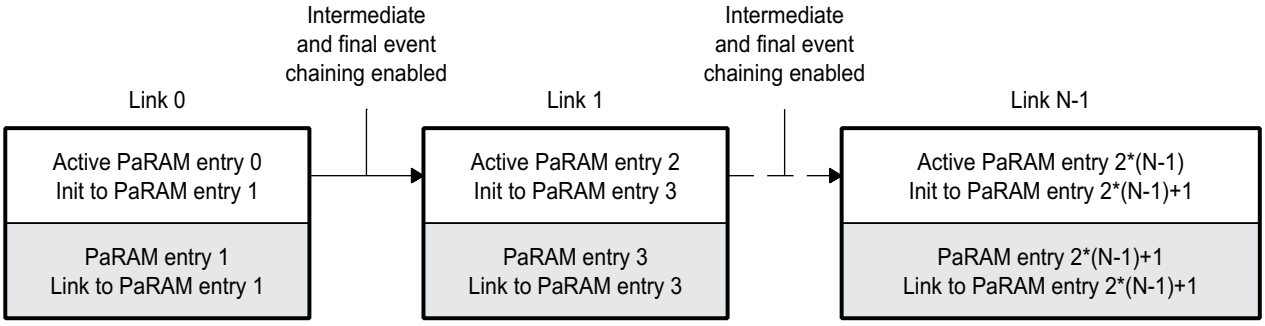
Parameter	Value	Description
ACNT	16	Number of bytes transferred in each array (each array consists of 8 chips, and 1 chip = 2 bytes on UL AxS)
BCNT	$2*M_0$	Over-sampling factor on UL × Num of AxS per link
CNT	4	Number of frames
SBIDX	16	B dimension increment for source in bytes
DBIDX	16	B dimension increment for destination in bytes
SCIDX	512	C dimension increment for source in bytes
DCIDX	1728	C dimension increment for destination in bytes
Intermediate chaining enabled		See Section 7.4.3.6.
Final chaining enabled		See Section 7.4.3.6.
Link update enabled	Yes	See Section 7.4.3.6.

7.4.3.6 EDMA Chaining and Link Updates

A single PaRAM entry is necessary to transfer eight chips for all active AxSs per link. Each link must be configured with its own PaRAM entry. An frame sync system event must be configured for each link. This system event triggers a specific PaRAM entry to perform the transfer.

If two or more links have the same timing on the inbound, then you can chain the EDMA entries so that only a single system event is required to trigger the necessary transfers. In this case, you must enable intermediate and final chaining.

Figure 75. Scheme Showing Event Chaining and Link Updates for Transferring Data From AIF Inbound RAM to RAC FE FIFO When Inbound Links Have Identical Inbound Timing



In addition to an active PaRAM entry, there must be a re-load entry. The re-load entry ensures that the re-load entry is copied into the active entry when the active PaRAM entry is exhausted. Link update must be enabled to support the re-load feature. Thus, two EDMA PaRAM entries are necessary per link to perform the data transfer from AIF to RAC.

7.4.3.7 EDMA PaRAM Entry Sizing

Table 60 shows the number of EDMA PaRAM entries that are required to transfer data from AIF inbound RAM to the RAC FE FIFO. One EDMA entry is required to perform the timestamp transfer from frame sync to RAC and two EDMA entries are required per link to perform the data transfer from AIF to RAC.

Table 60. EDMA PaRAM Entry Sizing for Transfer From AIF Inbound RAM to RAC FE FIFO

Description	Number
Number of PaRAM entries required to transfer timestamp	1
Number of PaRAM entries per link	2
Number of PaRAM entries per inbound link	3

7.4.3.8 EDMA Event Sizing

Each link must have its own DMA channel (and event) associated with the EDMA of data from AIF memory to RAC.

Table 61. EDMA Entry Event Sizing for Transfer From AIF Inbound RAM to RAC FE FIFO

Number of EDMA channels per inbound link	1

7.4.3.9 Frame Sync Event Sizing

You must configure the frame sync event generators so that the RAC is driven correctly. Table 7 shows the number of event required to operate the RAC.

Table 62. Frame Sync Event Sizing for Inbound Transfer From AIF Inbound RAM to RAC FE FIFO

Description	Number
Event to start RAC processing	1
Number of PaRAM entries per link	1
Event to AIF (frame sync tick)	1
Events to EDMA per link	1
Maximum number of event generator circuits required for RAC operation per inbound link	4

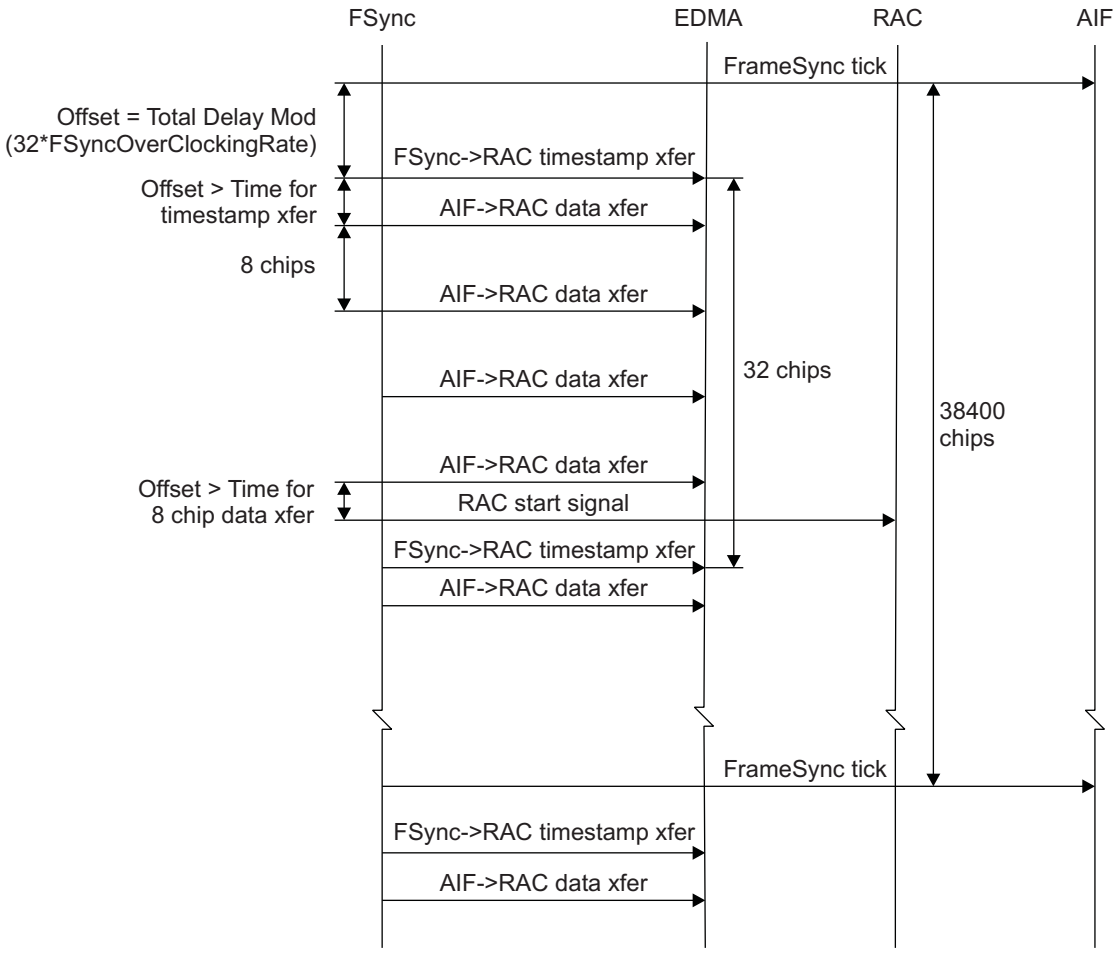
7.4.3.10 RAC Event Timing Relationships

The event timing relationships for RAC operation are shown in Figure 76. The event offset can be chosen to minimize the closed-loop latency in the system.

Consider these points:

- The system event triggering the EDMA of the timestamp from frame sync to RAC should be offset so that offset = (total delay) mod (32 × frame sync over-clocking rate), where 32 is the size of the AIF inbound buffer for each over-sampled AxS, frame sync over-clocking rate is the input clock rate to the frame sync divided by the chip rate (frame sync over-clocking rate = frame sync clock rate ÷ 3.84 MChips/sec), and the total delay is the latency of the UL chain through the system. If the TMS320TCI6487/8 units are in a daisy chain configuration, the total delay reflects the latency of the AxS from the point of origin to the time it gets written to the inbound AIF RAM.
- The system event triggering the EDMA data transfer from AIF to RAC should only occur when the time stamp transfer is complete. The offset ensures that the time stamp transfer is not broken by any EDMA transfer occurring simultaneously. The EDMA of data is periodic, occurring once every eight chips.
- The system event that triggers the start of the RAC operation should only arrive when the last eight chips (of the total 32 chips) arrives in the RAC. This can be ensured by ensuring that the offset between the EDMA data transfer system event and the RAC start event is greater or equal to the time it takes to complete the eight-chip EDMA transfer for all of the AxSs for the link.

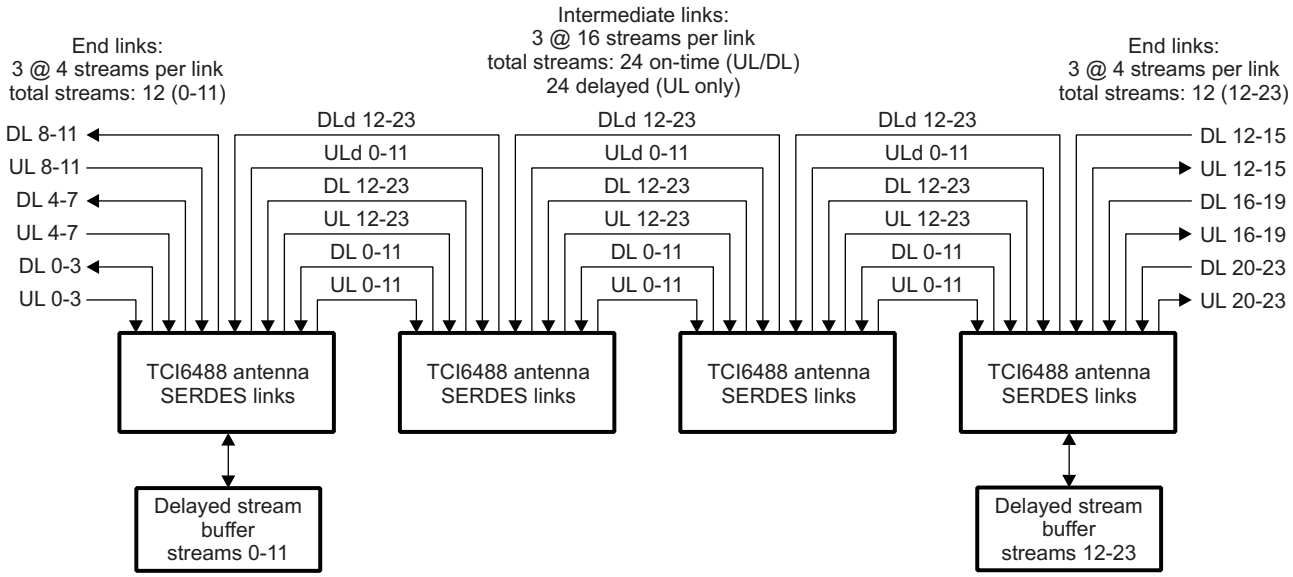
Figure 76. Event Timing Relationships Shown for Inbound Transfer from AIF Inbound RAM to RAC FE FIFO for a Single Link



7.4.4 Software Configuration for Delayed Data Streams

You can delay the UL antenna streams by a fixed latency so that you can determine the spreading factor by decoding the DPCCH. Once the spreading factor is known, it is used to de-spread the DPDCH using a delayed version of the same UL antenna streams. The use of delayed streams removes the need for buffering at the DSP. Figure 77 shows delayed streams generation for UL streams 0-11 and 12-23 at the end TMS320TCI6487/8 units. This scheme ensures that the intermediate TMS320TCI6487/8 get on-time and delayed versions of the UL on their inbound links, thus avoiding the need to buffer delayed streams. In addition to supplying the intermediate TMS320TCI6487/8 with delayed streams, the end TMS320TCI6487/8 can also locally source their RAC's with delayed streams. The set up for performing UL operation using RAC and generating delayed streams is shown in Figure 77.

Figure 77. Delayed Stream Generation Using Local Memory at End Devices



7.4.4.1 Configuration

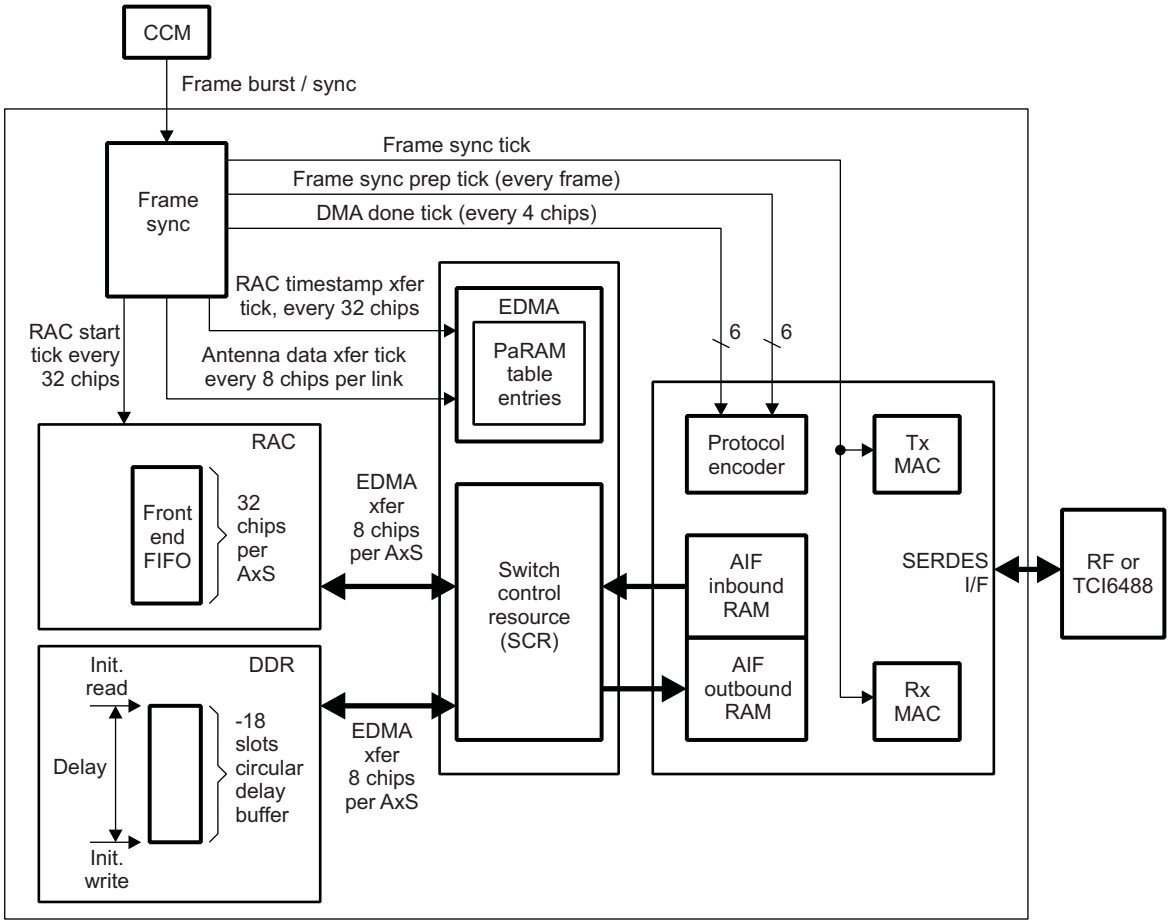
The set up for the performing UL operations is similar to the scenario discussed in Section 7.4.3. Since the delayed stream is generated and transmitted on an outbound link(s), the set up is similar for the outbound set up discussed in Section 7.4.1. The setup is discussed briefly in the following sections.

7.4.4.2 UL Operation (Inbound AIF RAM / DDR → RAC)

The frame sync event generator is set up to provide the AIF with the frame sync tick occurring every 38,400 chips (10msecs). The frame sync is also programmed to generate two system events to the EDMA: one of these events occurs every 32 chips and the second system events occurs every eight chips (per link). The system event occurring every eight chips is used to trigger the DMA transfer of on-time streams (inbound AIF RAM → RAC) and delayed streams (DDR → RAC) to the RAC.

The frame sync is also programmed to generate a system event to the RAC occurring every 32 chips.

Figure 78. Set Up for Generating Delayed Streams and UL Processing Using RAC



7.4.4.3 *Delayed Streams Generation (DDR → AIF Outbound RAM / RAC)*

In addition to the frame sync event generators programmed as described above, the frame sync is programmed to generate two system events to the AIF (per link) : one for the DMA done tick occurring every four chips and the second event is the frame sync prep tick occurring every 38,400 chips. In order to perform EDMA transfers, the same system event to the EDMA that occurs every eight chips is used to trigger the AIF inbound RAM → DDR and DDR → AIF outbound RAM transfers.

7.4.4.4 *Operation*

The operation of the RAC is similar to that described in Section 7.4.4.4. In this scenario, the RAC receives on-time and delayed streams. The application configures the RAC to handle the on-time and delayed streams.

1. EDMA transfers a time stamp from the frame sync to the RAC every 32 chips.
2. The time stamp transfer is followed by four EDMA transfers of data from AIF inbound RAM and DDR to RAC front end FIFO. Each transfer occurs every eight chips and transfers eight chips of I/Q data for up to 54 over-sampled antenna streams (over-sampling factor = 2). The antenna streams transferred from AIF inbound RAM to RAC constitute the on-time streams; the antenna streams transferred from DDR to RAC constitute the delayed streams. As explained later, the DMA transfers of on-time and delayed streams are chained EDMA events that are triggered when the EDMA receives a single system event occurring every eight chips.
3. The frame sync provides a system event to start the RAC operation when the EDMA transfer is complete. This system event occurs every 32 chips and must arrive at the RAC when the EDMA transfers for the 32 chips are complete.

The delayed streams are generated as follows:

1. The eight-chip tick to the EDMA is used to transfer on-time antenna streams from AIF inbound RAM to DDR and delayed streams from DDR to outbound AIF RAM. The transfer of on-time streams and delayed streams to the DDR and AIF is triggered using the same system event (eight-chip tick) to the EDMA. Since each inbound link has its own timing, a single system event is required per link to perform the on-time and delayed stream transfers. The outbound timing for the link containing the delayed streams depends on the latency involved in performing the DMA transfers in and out of the DDR and the latency required to decode the TFCI bits in the UL DPCCH.
2. The on-time streams read from AIF inbound RAM are written into a buffer in the DDR. The data is required to be transferred to the AIF outbound RAM after the desired latency. For this purpose, the EDMA PaRAM entries must be programmed so that the read and write offsets in the DDR will equal the desired delay for generating delayed streams. The amount of buffering required in DDR per link is equal to the desired latency. The size of the delayed stream buffer is approximately 18 slots in Figure 78.

7.4.4.5 Set Up Steps

The subsections below explain the steps to set up AIF, RAC, frame sync and DDR2. The order you use to set up these actions (subsections) is irrelevant. The setup for these modules should be performed before any UL processing happens to prevent any loss of data. The setup actions can be performed by any one of the three GEM cores on the TMS320TCI6487/8.

7.4.4.5.1 RAC Set Up Actions

The following steps are used to set up RAC:

1. Zero out the contents of the RAC input FIFO.
2. Set up the EDMA PaRAM entry to perform time stamp transfer from the frame sync memory-mapped register to the RAC register.
3. Configure and enable the frame sync event generator to generate a system event to the EDMA occurring every 32 chips for the time stamp transfer.
4. Configure and enable an frame sync event generator to provide a start signal to the RAC every 32 chips.

7.4.4.5.2 Frame Sync Set Up Actions

Set up the RP3 and/or system timer in frame sync so that system events can generate.

7.4.4.5.2.1 AIF Set Up Actions

The following steps are used to set up AIF:

1. Zero out the inbound memory
2. Set up the frame sync tick so that a system event generate every 10msecs. The Tx MAC and/or Rx MAC in AIF uses this tick to synchronize with external links. You can skip this step if this is already set up.
3. Perform the following steps for every link that is set up:
   a. Configure a PaRAM entry so that the EDMA can perform on-time antenna data transfers from AIF inbound RAM to RAC.
   b. Configure and enable a frame sync event generator to cause a system event to the EDMA occurring every eight chips to perform on-time and delayed stream data transfers.
   c. Configure PaRAM entries to generate delayed streams. These transfers include AIF inbound RAM → DDR, DDR → AIF outbound RAM, AIF outbound RAM → RAC. The delayed stream transfers are chained to the on-time transfer (AIF inbound RAM → RAC).

In step 3 above, the EDMA set up must precede the frame sync event generator set up if the links are added dynamically.

7.4.4.5.2.2 DDR Set Up Actions

Set up the delayed buffer and zero out the contents. The size of the delay buffer is dependent on the latency required for delayed streams. The latency includes any transfer time required to transfer data to and from the DDR. The size of delay buffer should be a multiple of eight chips because during each transfer event eight chips of antenna data must be transferred.

The frame sync timers (RP3 and/or system timer) are enabled when the set up steps outlined above are completed. The enabling of timer(s) allows for system event generation and periodic actions listed in Section 7.4.4.4 to be performed.

7.4.4.6 Memory Organization

There are four data transfers for the generation of delayed streams and operating the RAC.

These are described below:

- AIF inbound RAM → RAC
- AIF inbound RAM → DDR2
- DDR2 → AIF outbound RAM
- AIF outbound RAM → RAC.

  The memory organization for AIF inbound RAM → RAC data transfer is discussed in Figure 80 and the EDMA transfer parameters are shown in Table 63. The memory organization and transfer parameters required for performing the AIF outbound RAM → RAC is similar to that shown in Figure 79 and Table 66, respectively except that the source and destination parameters are now reversed.

  The memory organization for DDR2 and AIF inbound / outbound RAM is shown in Figure 79. The AIF stores UL data in the same manner; hence, the memory organization shown in Figure 79 applies to inbound and outbound AIF RAM.

7.4.4.6.1 Delayed Stream Example Assumptions

In order to demonstrate memory organization for delayed stream generation, it is necessary to make a few assumptions.

The assumptions made concerning the parameters affecting EDMA parameters are listed in Table 63.

Table 63. Assumptions Made to Demonstrate Delayed Stream Generation

Description	Amount
Number of chips per EDMA transfer	8 chips
Number of bytes per chip	2
AxS over-sampling factor	2
Number of Antenna Streams for Link 0	M_0
Size of delay buffer in DDR2	46080 chips (18 slots)

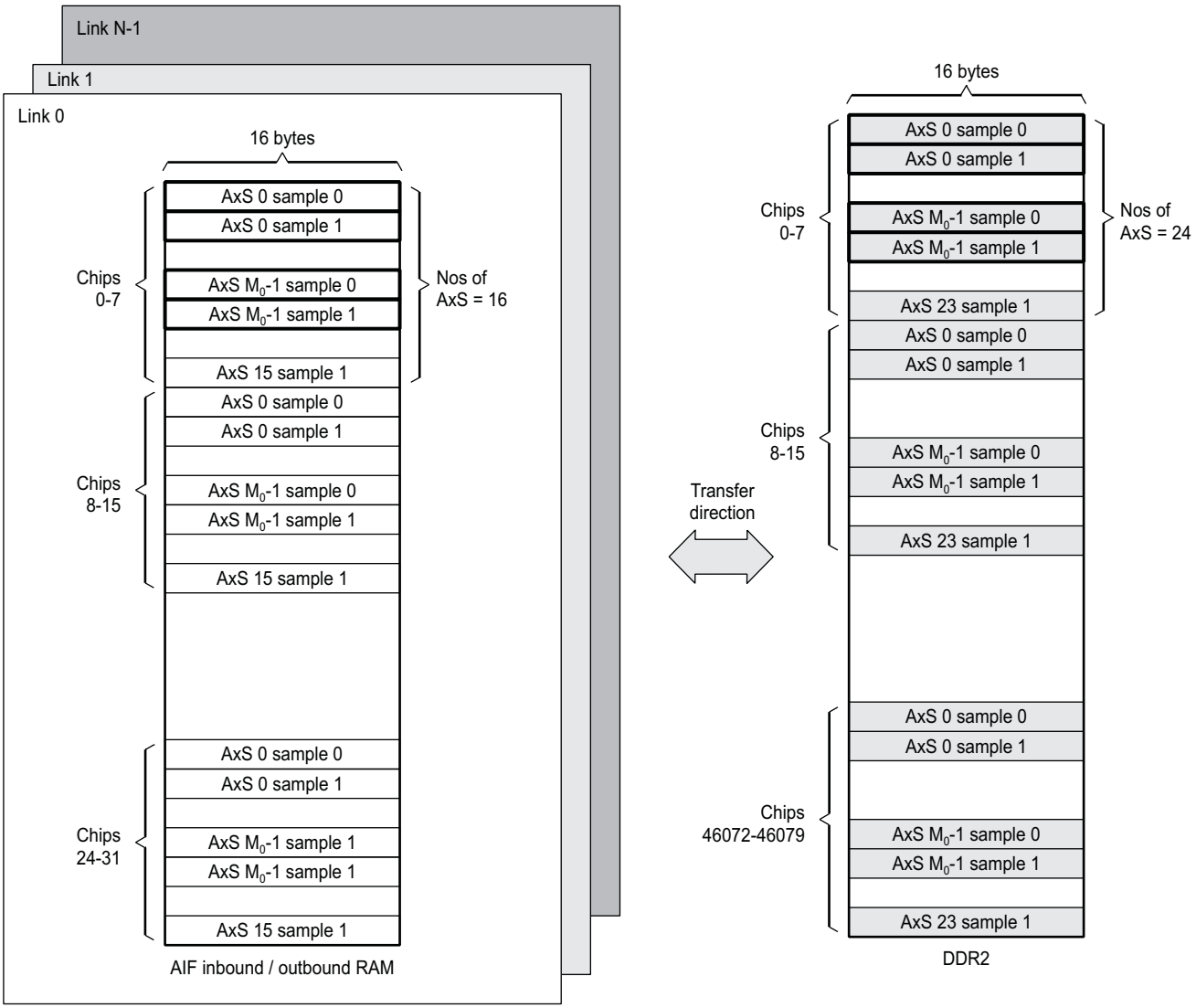
7.4.4.7 AIF RAM

The AIF RAM stores UL data in groups of eight chips. Even and odd samples of the same AxS are stored continuously. The AIF inbound and outbound RAM store UL data in a similar manner. Although the data format in the AIF outbound buffer is not the same as the OBSAI or CPRI formats, as long as the data format is configured for UL RSA format the AIF packs the data correctly in the outbound SERDES link.

7.4.4.8 DDR2

A circular buffer is configured in DDR2 which can hold up to 24 antenna stream and up to 18 slots (46,080 chips) of I/Q data. Each over-sampled antenna stream is arranged in eight-chip blocks of 16 bytes (one chip = two bytes), as shown in Figure 79. The total memory required to hold 24 antenna streams is 24×46080×2×2 = 4,423,680 bytes.

Figure 79. Memory Organization in DDR2 and AIF Inbound/Outbound RAM



7.4.4.9 Transfer Parameters

Table 64 shows the transfer of UL data for a single link from DDR2 → AIF outbound RAM. An AB sync transfer is used to perform the transfer. The figures for EDMA transfer are based on assumptions listed in Table 64.

The EDMA is triggered every eight chips to perform the data transfer from the AIF inbound RAM to DDR2. Data that is transferred in the first EDMA transfer on receiving a system event from the frame sync is shown in dark lines in Figure 80.

The important PaRAM entries for performing the EDMA are listed in Figure 80.

Figure 80. DMA Shown for Inbound AIF RAM → DDR2 Transfer for a Single Link

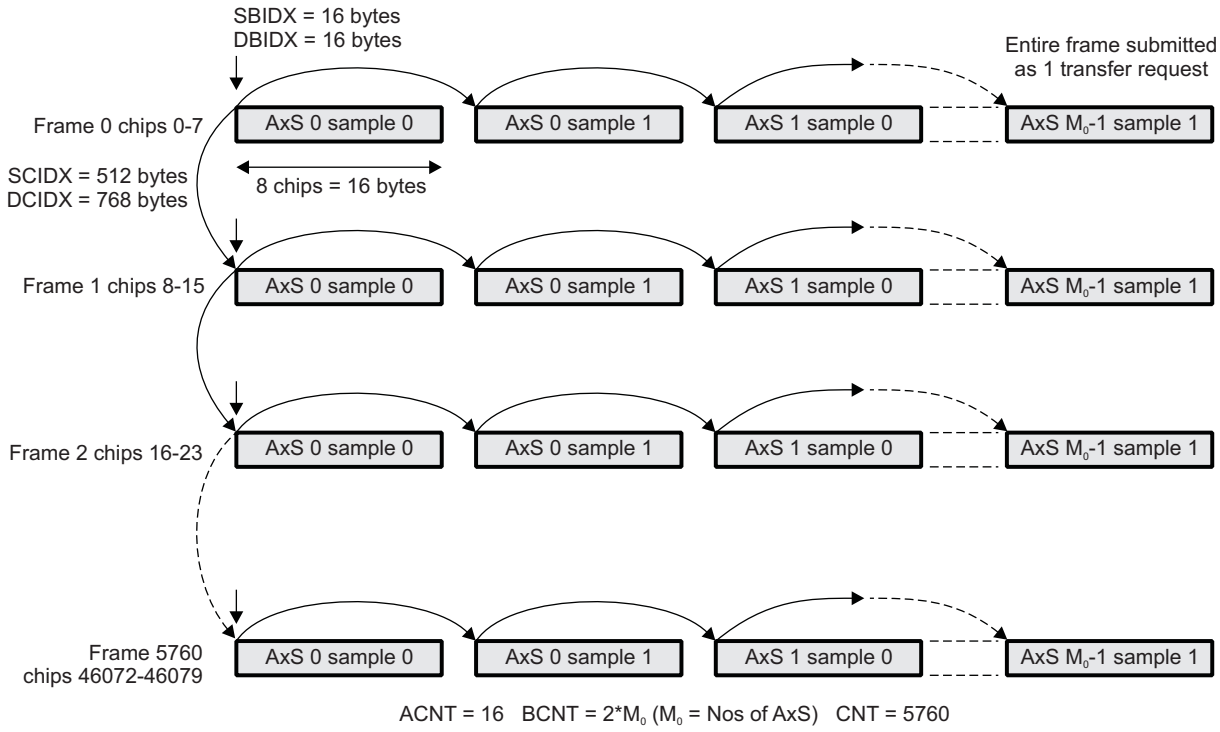


Table 64. EDMA PaRAM entries for AIF inbound RAM → DDR2 transfer

Parameter	Value	Description
ACNT	16	Number of bytes transferred in each array (each array consists of 8 chips, and 1 chip = 2 bytes on UL AxS)
BCNT	$2*M_0$	Over-sampling factor on UL * Num of AxS per link
CNT	4	Number of frames
SBIDX	16	B dimension increment for source in bytes
DBIDX	16	B dimension increment for destination in bytes
SCIDX	512	C dimension increment for source in bytes
DCIDX	768	C dimension increment for destination in bytes
Intermediate chaining enabled		See Section 7.4.4.10.
Final chaining enabled		See Section 7.4.4.10.
Link update enabled	Yes	See Section 7.4.4.10.

The transfer from DDR2 → AIF outbound RAM uses similar entries to Figure 80, except with the source and destination are reversed.

7.4.4.10 EDMA Chaining and Link Updates

There are four PaRAM entries that are necessary to perform the scenario shown in Figure 78. The following data transfers are required:

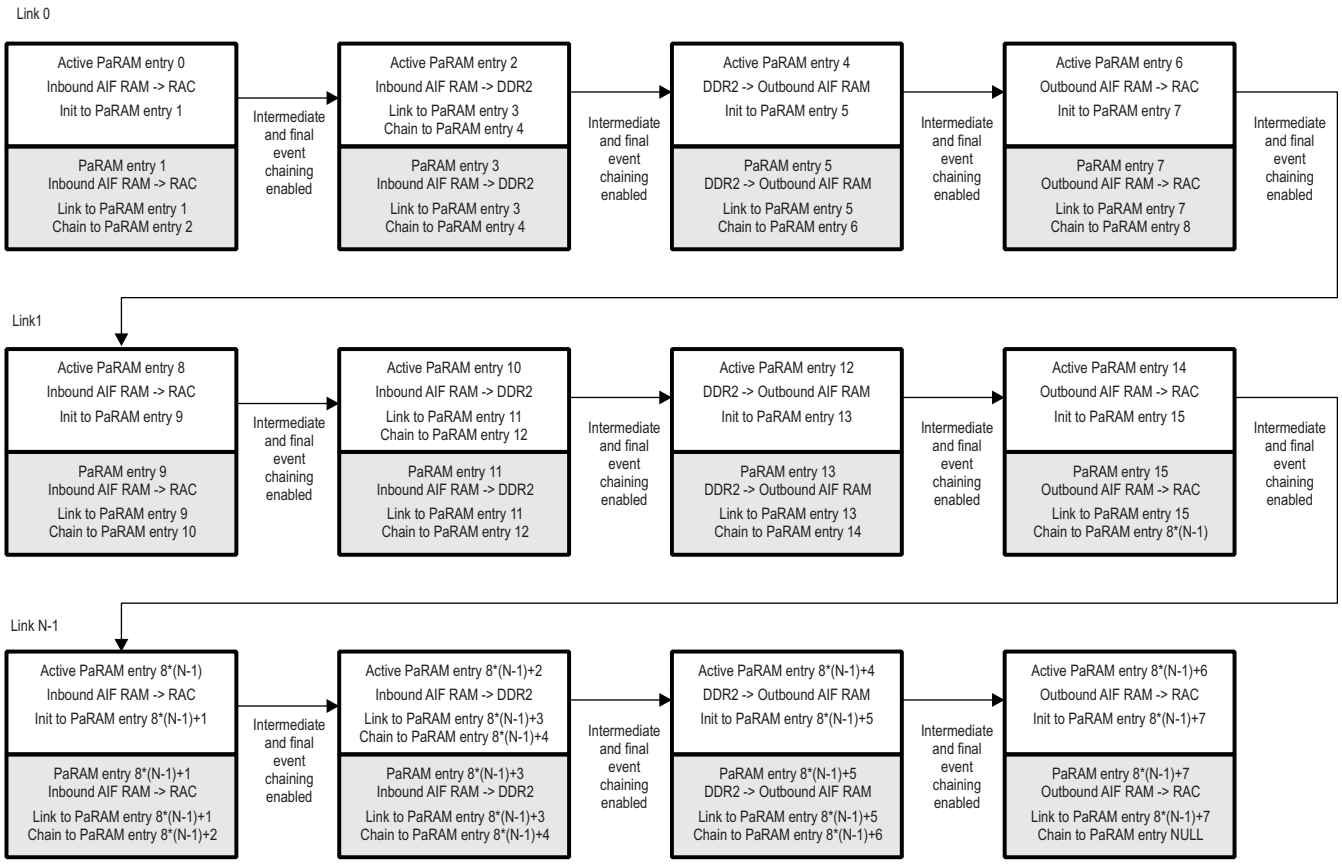
- Every inbound link containing on-time antenna streams requires the following transfers:
  - AIF inbound RAM → RAC
- Every outbound link containing delayed antenna streams requires the following transfers:
  - AIF inbound RAM → DDR2
  - DDR2 → AIF outbound RAM
  - AIF outbound RAM → RAC

    The AIF outbound RAM → RAC transfer is required because the RAC processes on-time and delayed streams in the example shown in Figure 78.

  The four PaRAMs must be chained so that a single system event from frame sync triggers all of the transfers. Intermediate and final chaining must be enabled.

Each PaRAM entry requires a re-load entry when the active entry is exhausted. Thus, eight PaRAM entries are necessary for each inbound and outbound link to perform the data transfer (see Figure 81). The re-load entries are identical to the active entry; however, the PaRAM entry that corresponds to the inbound AIF RAM → DDR2 transfer is an exception (see PaRAM entry 2, 10, .. in Figure 81). The initial write pointer is offset from the start of the delay buffer in DDR2 by the desired antenna stream delay. However, when the write reaches the end of the circular buffer, it wraps to the start of the buffer. Since the initial write location to DDR2 is different from the write location when the circular buffer wraps, the initial PaRAM entry is different from the re-load entry.

Figure 81. Scheme Showing Event Chaining and Link Updates for Performing UL Processing Using RAC and Generating Delayed Streams.



If the inbound and outbound links have the same time, then you can chain the PaRAM entries that belong to different links, as shown in Figure 81.

7.4.4.11 EDMA PaRAM Entry Sizing

Table 65 shows the number of EDMA PaRAM entries that are required to perform the data transfers for the scenario shown in Figure 78. One EDMA entry is necessary in order to perform the timestamp transfer from frame sync to RAC; and, eight EDMA entries are required for each inbound and outbound link in order to perform the data transfers. The inbound link carries the on-time uplink antenna streams and the outbound link carries the delayed streams.

Table 65. EDMA PaRAM Entry Sizing for Generating Delayed Streams and Operating the RAC

Description	Number
Number of PaRAM entries required to transfer timestamp	1
Number of PaRAM entries per inbound and outbound link	8

7.4.4.12 Frame Sync Event Sizing

Table 66 shows the number of events that are required to generate delayed streams and to operate the RAC.

A single EDMA event is necessary in order to perform the data transfers. If two or more links have the same timing, then you can use the same system event to EDMA to chain all of the necessary data transfers (see Figure 81).

Table 66. Frame Sync Event Sizing for Generating Delayed Streams and Operating the RAC

Description	Number
Event to start RAC processing	1
Event to transfer timestamp from frame sync to RAC	1
Event to AIF (frame sync tick)	1
Events to EDMA	1
Events to AIF per outbound link	2
Number of event generator circuits required to perform delayed stream operation	6

7.4.5 Using TCs to Minimize Queuing

When multiple channels are enabled and DMA transfers must happen simultaneously the queues might be exhausted and some event might miss. You can avoid this issue by mapping different channels into separate TCs. Mapping channels into separate TCs is highly essential when all six of the links are enabled.

7.5 Inter-TMS320TCI6487/8 Communication

Both OBSAI and CPRI support inter-TMS320TCI6487/8 communication. Control messages for OBSAI and vendor-specific control words for CPRI provide the capability for non-IQ data transfer. Since you can use the non-IQ data transfer for transferring any content, you can use this to pass data that you have defined between devices.

7.5.1 Options for OBSAI and CPRI

OBSAI offers more options for transferring non-IQ data compared with CPRI:

- OBSAI is more flexible since the address field in the header can target another TMS320TCI6487/8 (or multiple TMS320TCI6487/8). CPRI has no way of specifying the destination. Therefore, you must indicate that each TMS320TCI6487/8 can receive all of the control words and the DSP processing must parse the content of those words to determine whether or not the contents are relevant, and how to use them if they are for CPRI. This is a customer-specific implementation.

- OBSAI's transmission rules allow multiple TMS320TCI6487/8 to use one link. That is, one TMS320TCI6487/8 can transmit on even control words and a second TMS320TCI6487/8 can transmit on odd control words. The TMS320TCI6487/8 implementation has no limitations on transmission rules for control message slots (limitations for using data message slots do exist). CPRI does not have this capability.
- Each TMS320TCI6487/8 in a daisy chain can either not write to the vendor control words or it over-writes all of the control words. The only CPRI option for control-word sharing on a link is for each TMS320TCI6487/8 to receive all control words, change only the content assigned to them in local memory, and then send out the modified control words.

7.5.2 Using Standard Control Messages

Only OBSAI supports sending TMS320TCI6487/8-to-TMS320TCI6487/8 data using standard control messages. Packet-switched data can be targeted at data messages as well as control messages. The bandwidth is limited to what data rate can be supported to/from the packet-switched FIFOs. A minimum bandwidth of two AxC slots can be supported; with proper handling, the bandwidth can allow for many more AxC slots.

For example, this feature allows a 4x link with 16 AxC slots to use two AxC slots for non-IQ data and the other 14 slots for standard IQ data. Transmission rules allow multiple TMS320TCI6487/8 to use the bandwidth that is available on the two AxC slots, but there are some limitations to the modulo value. Since packet- switched data has software generate the header contents, the address field can target any other TMS320TCI6487/8.

7.5.3 Using Unused Links

If not all links are being used for IQ data, an entire link(s) can be used for non-IQ data passed between TMS320TCI6487/8 units, thus allowing for the maximum bandwidth. Both the OBSAI and the CPRI support this option. This mechanism uses the circuit-switched data path; therefore, data must be continuously sent. This mode limits transmission rules so that an TMS320TCI6487/8 cannot share AxC slots. For example, if a 4x link is used, each TMS320TCI6487/8 can transmit on one of the 16 AxC slots; but each AxC slot is only dedicated to one TMS320TCI6487/8.

Since the header is appended within the AIF for circuit-switched data, the address must be set up before hand; therefore, it is not as flexible as using packet- switched data. The TMS320TCI6487/8 units can transmit and receive entire links for CPRI, but there is no direct support for the purpose of sharing links between multiple TMS320TCI6487/8. It is possible to share a link using the aggregator function by having each TMS320TCI6487/8 put zeroes in AxC locations not assigned to them and a TMS320TCI6487/8 further down the daisy chain can aggregate its contents into that location.

7.6 Error Handling

7.6.1 Bit Errors

SERDES links have bit error rates (BER) with approximate frequencies of one error per day. A single SERDES bit error only causes frame sync failure if it affects the K28.7 character or two consecutive K28.5 characters. The probability of this occurrence is on the order of once per 400 years (per link).

Statistically, it is unlikely that messages will go missing. The only known (non-catastrophic) reason that a message will go missing is if a SERDES bit error corrupts the type or address field on a message. Statistically, this should only occur once per seven days per link. Even when this error condition occurs, it is unlikely that any calls will be dropped.

The Rx MAC is tolerant of some amount of bit errors occurring in a K28.5 IDLE byte. If one error occurs in isolation occurs an error is flagged and the Rx MAC inserts the byte. Two such errors in a row cause the link to shut down. Since K28.7 IDLE bytes occur with a low frequency, the probability of receiving a bit error in this IDLE byte only is very low; thus, the Rx MAC causes an error and the link shuts down.

7.6.2 Comma Alignment

The SERDES receiver examines the data stream for the positive or negative disparity comma characters (0011111xxx and 1100000xxx, respectively) when comma alignment is enabled. If you detect a comma that straddles symbol boundaries, you can adjust the relationship between the received byte clock and the running disparity to correct the alignment. This results in one symbol of corrupted data, as between 1 and 9 bits are removed from the data stream.

Comma alignment can be either enabled or disabled in the receive part of SERDES:

Example:

pSerdesSetup[0]. BEnableRxAlign = TRUE;

Note: When a link goes down, the recovered receive clock continues to operate at the rate before the link went down, but is unreliable due to the effects of noise on the link.

You should disable comma alignment in OBSAI mode when Rx MAC achieves frame synchronization.

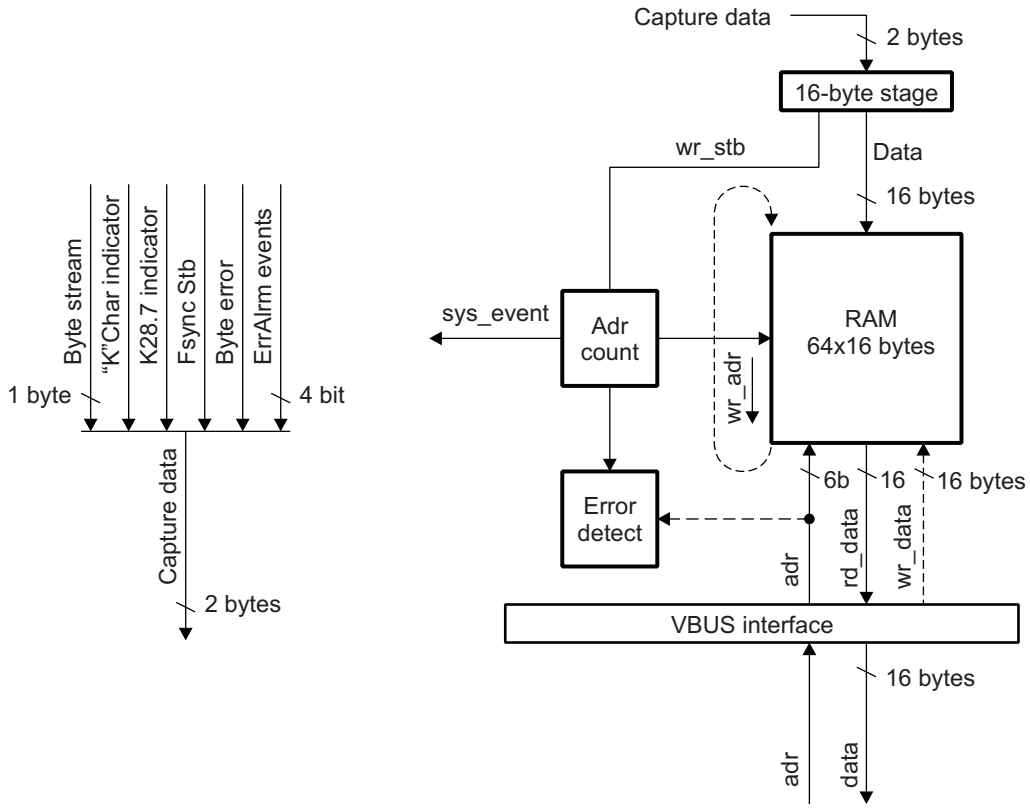
7.7 Debug Capabilities

7.7.1 Message Capture on Event

When error/alarm conditions are triggered, the contents of the link can be captured in a data capture buffer. The data trace buffer is a circular RAM structure that is used to capture whole link byte streams. The system uses the capture buffer in conjunction with the error/alarm error conditions to emulate an internal logic analyzer function. The capture buffer is a circular RAM structure that the Rx MAC circuits feeds. You can write the contents of a single SERDES link to the capture buffer. Internally, captured link data will be DMA'ed from the AIF to internal or external bulk RAM.

7.7.1.1 Data Capture Buffer

Figure 82. Data Capture Buffer



The data trace capture mechanism used in conjunction with the TPDMA and external RAM captures a link's byte stream for offline analysis. The AIF concatenates the link byte stream with other diagnostic signals and stores the data into a circular buffer located in the AIF data buffer as shown in Figure 82. The AIF periodically triggers the TPDMA to read a burst of captured data from the data capture buffer RAM and writes the values to the external RAM.

A byte stream and other diagnostic signals from a single link are passed to the data capture buffer RAM from the Rx MAC. Internally, the Rx MAC has selected one of the six links that should be routed to the data capture buffer for potential capture. The data capture buffer has additional MMR bits that either enable or disable the data capture function.

Table 67. Data Buffer Capture Trace Word

# Bits	Data Field	Description
1	K28.7	Set if the data value is the K28.7 code.
1	frame sync strobe	This is the captured form of the frame sync strobe signal.
1	byte error	Set if the 8b10b decoder indicates a line code violation.
1	K character	Set if the data value is a K character code.
4	system events	Indicator signals which mirror activation of the four Error/Alarm system events.
8	byte stream	Raw byte stream of data exiting the 8b10b decoder

The byte stream is concatenated with diagnostic signals resulting in a two-byte capture, as shown in Table 67. The K character indicator identifies the current byte as a control K character, as opposed to a data byte. As a matter of visual convenience, the K28.7 character is specifically identified by the data capture buffer. This field is set any time the K character field and data byte indicates a K28.7 character. The byte error indicator shows that an 8b10b error occurred during the decoding of the current byte. These signals are sourced from the 8b10b decoder in the RX_MAC.

Other captured signals of interest are error/alarm system events and the internal frame sync strobe signal that the frame sync module supplies. These signals may not coincide exactly with the appropriate byte stream due to the latencies of the independent strobes. Therefore, these signals are edge- detected, registered, and stored with exactly one associated data byte.

The 16-byte staging register stores eight two byte half words of captured data prior to writing into the RAM. Once 16 bytes of data are stored, the entire quadword is written into the data capture buffer RAM.

For debug purposes, the VBUS may both read and write to the data capture buffer RAM. Reads are for normal operation, but writes are for debug operation only. VBUS writes might contend with data capture buffer writes and are given a lower priority; VBUS stalls result.

The data capture buffer RAM is two-ported RAM of 64 addresses x 16 bytes wide. Both halves of the address space are treated as ping/pong accesses. While the capture is writing to one half of the RAM, the VBUS should read the other half of the RAM. The DMA access rate is determined by the size of the RAM along with the maximum OBSAI RP3 byte rate. The TPDMA transfer 512 bytes should be triggered by the dedicated system event.

The system event trigger rate is:

- 1.2 MHz for a 4x link rate
- 600 KHz for a 2x link rate
- 300 KHz for a 1x link rate

Calculations for 1x link:

300 KHz = 768Mbps x 0.8 (8b10b) x 1/8 (bytes/bit) x 1/8 (storage of 16 bytes) x 1/32 addresses

Data capture buffer configuration involves enabling the data capture buffers by setting CAPTURE_EN bit in DB_GENERIC_CFG register (see Section 8.2.13.1). For example:

```
arg = TRUE;
CSL_AifHwControl (hAifLink, CSL_AIF_CMD_DB_ENABLE_LINK_CAPTURE, &arg);
```

7.7.2 Logic Analyzer Capabilities

In order to achieve better visibility into the antenna interface external traffic; data tracing is available. Data tracing allows for faster verification and testing. Data tracing is comprised of a data capture mechanism and a triggering mechanism. You can use these separately; but, create a built-in logic analyzer functionality when they are used together.

The basic features of data tracing are:

- Raw data from a selected link can be transferred to either an external memory or to the L2 memory of one of the DSP cores.
- The antenna interface can generate up to four events to the GEM emulation logic.
- The GEM emulation logic sets up state machines and triggers the system.

Triggering is implemented with a combination of the four antenna interface system events and the advanced event triggering (AET) capabilities of the GEM core. All four of the antenna interface system events are inputs to all of the GEM cores.

Each of the four system events can be programmed for any of the error/alarm conditions. Since each condition can be enabled/disabled individually for each AI_EVENT output, this offers an OR function for each event. Conditions are normally registered when asserted and cleared by software. The antenna interface offers the capability to have each condition to automatically clear. This would allow, for example, conditions to be asserted and cleared multiple times without software intervention. This could be used for counting events or stepping through a state machine.

Each error or alarm event is reported to the exception event handler (EE) from its source module (Tx MAC, Rx MAC, etc.) and results in a unique bit being set to a 1 in an interrupt source raw status register (see Section 8.3.7.2) in the EE. This bit remains set until cleared by the DSP through the VBUSP configuration bus. All error or alarm events can be routed to any of the AI_EVENT [1:0] signals, and you can configure the subset of error or alarm events that are associated with each of the AI_EVENT [1:0]

signals through individual mask registers for each of the events inside the EE. If the error or alarm event is enabled through the mask, it triggers the AI_EVENT[1:0] it is routed to when the event occurs. After a particular AI_EVENT[1:0] is triggered, subsequent error or alarm events that occur for this same AI_EVENT[1:0] do not trigger the event again until the DSP has acknowledged the interrupt by writing to the EE end of interrupt register (see Section 8.2.14.4).

The following example configures AI_EVENT0 for the master frame boundary range error. For details regarding the exception event handler register, see Section 8.3.7.1.

```
evntLinkEn.excEvtType = CSL_AIF_ERR_EVENT_0;
evntLinkEn.excMaskA = 0x80;
evntLinkEn.excMaskB = 0;
CSL_aifHwControl (hAifLink[0], CSL_AIF_CMD_EE_ENABLE_LINK_MASK, &evntLinkEn);
```

The enabling and disabling of the data capture occurs when the software writes to the data capture enable register. The exact timing of this access is not predictable due to the variable latencies in the TPDMA. Therefore, it is possible to halt the capturing of data to the circular buffer without halting either the AIF or the software.

Features of AET:

- Supports a state machine with up to four state levels
- Supports a combinatorial logic look-up table (LUT) for up to five inputs
- Supports up to four counters/timers
- Some of the supported trigger resources
  - External events (external from the GEM core)
    - The four system events from the antenna interface
    - A GPIO pin (for an external trigger source)
    - Any other event available to the GEM cores
  - Counters (up to four)
    - Watermark mode for counters/timers is supported
  - State Sequencers (creates a maximum of four state events)
  - Other generated triggers (up to four triggers can be generated)
- Some of the triggers that can be generated are:
  - Program halt - halt program flow
  - Counter count – decrement counter or start timer
  - Counter reload - reload counter or stop timer
  - State change - change state machine or sequencer state
  - External triggers - trigger at chip boundary

Note: The AET uses an element called a trigger builder to associate input conditions with triggers to be generated.

Examples of external triggers:

- Delayed trigger
  - AIF Configuration
    - Configure AIF to generate AI_EVENT0 when an LOS is detected by the Rx MAC.
  - AET Configuration
    - Configure one of the four counter/timers (i.e. timer0) to generate a trigger when the desired time duration has passed.
    - Configure the trigger builder #1to generate a counter count trigger to timer0 when AI_EVENT0 is asserted.
    - Configure trigger builder #2 to generate a program halt trigger when the timer0 trigger is generated.
  - Operation: When an LOS occurs on the designated link AI_EVENT0 is generated. Trigger builder 1 starts timer0. When timer0 reaches the programmed threshold it generates an event. This event causes Trigger Builder #2 to issue a Program Halt trigger which will issue a system suspend signal.

This signal will cause the TPDMA to stop DMA transfers and the AIF to go to its suspend state.

- State machine-based trigger
  - AIF Configuration
    - Configure AI_EVENT0 to be generated when the link finds a type field with the value 0b00100.
    - Configure AI_EVENT1 to be generated by each K28.7 code.
  - AET Configuration
    - Configure trigger builder #1 to generate a state change trigger to move the state machine from state0 to state1 when AI_EVENT0 occurs.
    - Configure trigger builder #2 to generate a counter count trigger to counter0 whenever AI_EVENT1 is asserted and state = state1.
    - Configure trigger builder #3 to generate a system halt trigger when Counter0 = 10.
  - Operation: When the Type field = 0b00100, wait for 10 frames and then trigger.

7.7.3 Emulation Modes

The antenna interface supports emulation. The control of emulation is supported with the emulation control register and the hand shaking of two signals: EMUSUSP and EMUSUSP_RT.

The emulation control register supports RT_SEL and FREERUN bit fields. The SOFT bit field is not supported.

In emulation mode, the antenna interface allows daisy chain data to flow through the device, but data extraction, insertion, and aggregation are disabled in emulation mode. the Protocol Decoder and Protocol Encoder operations cease with the assertion of either EMUSUSP or EMUSUSP_RT (qualified by FREERUN and RT_SEL) inside the AIF. If data tracing is enabled, capture into the data trace buffer is halted. In addition to halting the PE, PD, and data trace capture operation, the error/alarm registers do not allow new error conditions to change the state of registers, allowing the DSP cores to observe the error/alarm state at the moment the emulation suspend state went active. The VBUS interfaces do not halt and they allow normal VBUS read/write activity during the emulation suspend state. All RAM and memory-mapped registers are accessible by the DSP cores for normal access including the data trace buffer.

When the above conditions are true, the AIF does the following:

- Disables all events that the AIF generates.
- Switches to pass-through mode
  - SERDES, Rx MAC, combiner/decombiner and Tx MAC operate like normal
  - Aggregator passes through the combiner/decombiner data (disregards protocol encoder inputs). Additionally, if you configure the aggregator to only pass the protocol encoder data, it stops sending data in emulation mode. This is also true if the combiner/decombiner loses synchronization while in emulation suspend mode.
  - Protocol decoder and protocol encoder freeze operation
  - Data buffer operates normally accepting VBUS reads and writes
- Hold the memory contents
  - Error alarm rejects new error conditions, freezing state at which halt activated
  - Protocol encoder and protocol decoder freezes all status for DSP inspection
  - Data buffer is available for DSP observation

When the emulation suspend state is deactivated, the LPSC must reset the AIF module.

Note: The process of entering and exiting the emulation mode causes several error or alarm conditions in most cases. Eventually, the AIF flushes its pipelines and re-synchronizes timing. The recovery process can take several UMTS frames.

7.8 Special Cases

7.8.1 AIF/Frame Sync Operation During DSP Reset (Except PORz)

The AIF is not reset by any DSP resets except for the PORz signal. The operation actions during other resets are:

- AIF and frame sync registers are not affected, so the frame sync continues to generate events, AIF continues processing inbound / outbound links.
- On inbound links, the AIF continues writing data into its inbound RAM; however, EDMA transfers to the DSP do not occur since the EDMA channels are disabled during reset. However, inbound links that are redirected or combined to outbound are not affected by the DSP reset. Hence, any daisy chained link is unaffected by a DSP reset if there is no DSP contribution to that link.
- On outbound links for which the DSP is contributing data, since the DSP is in reset, EDMA transfers from DSP to AIF do not return correct data. Thus, data on the outbound link is dirty. In order to avoid sending dirty data, the application may do one or both of the following:
  - Zero the outbound AIF RAM for the contributing link
  - Change the aggregator mode for the link to no operation. The aggregator mode is set to pre-reset value after the DSP comes out of reset
- After the DSP comes out of reset, it can continue to process inbound/outbound data as before. The DSP must re-enable and hook up events to their pre-reset state to process events from AIF and other peripherals.
- EDMA memory-map registers are cleared after a DSP reset. Thus, the EDMA must be configured to its pre-reset state. The PaRAM entries are not cleared during a reset. The application is required to enable events for those channels that do the inbound/outbound transfers. The time at which these channels must be enabled must be accurate since the channels are out of sync with frame sync events.

7.8.2 AIF/Frame Sync Operation During Frame Sync Re-Synchronization

During frame sync re-synchronization, the AIF and frame sync operation actions are:

- The Fsync re-synchronizes to external RP1 frame burst if the re-synchronization has been set. Fsync re-synchronization causes events to generate with different timing than prior to resync. The resynchronization may cause events to shift by a couple of subchips or many chips. The frame sync starts generating events from a frame boundary after resynchronization has occurred.
- Since the latencies in the Tx and Rx chains have not changed, the pi and delta values for Rx MAC and Tx MAC do not need to change. However, since Tx MAC and Rx MAC have individual state machines, it can take a couple of frames for the Rx MAC or Tx MAC to synchronize with the new timing; during this time the empty data is transmitted on outbound and no data is decoded on inbound. Therefore, an frame sync re-synchronization causes the disruption of links in the daisy-chain topology.
- EDMA has no re-synchronization procedure. PaRAM entries are not re-loaded when resynchronization occurs; the first event that arrives after resynchronization continues as the last event prior to reset. This results in EDMA being out-of-phase with frame sync events.

  One way to synchronization EDMA PaRAM entries with frame sync events is to have a disabled frame sync perform initialization of active PaRAM entries in the EDMA and re-enable the frame sync timer. This solution causes EDMA events to be in synchronization with PaRAM entries once event generation from frame sync resumes. The problem is that disabling and enabling the frame sync timer causes the frame sync to resynchronize only when a sync burst arrives. Since sync bursts periodicity may be large (> 100 frames), it can take time to start generating events again.

7.8.3 AIF Link Reconfiguration

The following steps should be followed:

1. It is strongly advised that you disable a link before any link re-configuration (you can reconfigure link properties in enable state, but output may be uncertain during transition). You can disable inbound/outbound links independently if there is no dependency in the paths (no combining or aggregation operation).
2. Reconfigure any inbound link using CSL API call CSL_aifHwSetup() for the new link configuration and enable the Rx link. You can modify certain properties using CSL API call CSL_aifHwControl().
3. Reconfiguration involving modifying values that impact more than one link requires disabling all links that are impacted due to the change. For example, if the reconfiguration of the inbound link involves modifying the type values that are contained within the protocol decoder type LUT, you should disable all of the inbound links before the change is made.

7.8.4 Loss of Sync Handling (Due to Link Failure or Hot Swap)

There are two reasons for link failure:

- Partial failure of the link due to the frame boundary misalignment
- Total failure of the link: this may be either due to the hardware providing the link data failing or because it has been removed (i.e., hot swapped)

  If a partial link failure occurs, AIF automatically attempts to re-synchronize. This is explained in sections Section 7.3.3.1 and Section 7.3.4.1.

  It is only a total link failure if the particular link is affected. All other links are not affected. If that link is an input to a combined link, the AIF inserts empty messages at the appropriate places until that link has re-synchronized. To recover from a complete link failure, you should follow the same steps as described in the link synchronization sections Section 7.3.3.1 and Section 7.3.4.1 only for that failed link. SERDES is electrically tolerant of hot-swapping (but may lose recovered clock). The AIF must be tolerant to hot swapping requirements. Logic in the AIF tolerates this condition. Also, the TMS320TCI6487/8 must not be damaged by the hot swapping. For further physical discussion on this topic, refer to the *TMS320TCI6487/8 Hardware Design Guide*.

7.9 Overview of CSL

The chip support layer constitutes a set of well-defined API that abstracts low-level details of the underlying SoC device so that you can configure, control (start/stop etc.), and have read/write access to peripherals without worrying about register bit-field details.

The CSL is organized into modules by peripheral. Each module contains a twin-layer user interface: the register layer and the functional layer.

The register layer header file for the AIF is provided in CSLR_AIF.H. The functional layer header file for the AIF is provided in CSL_AIF.H.

CSL_AIFINIT(), CSL_AIFOPEN(), CSL_AIFHWSETUP(), CSL_AIFHWCONTROL() AND CSL_AIFGETHWSTATUS() are the only APIs used in configuring the AIF. (For descriptions, see *TMS320TCI6487/8 Chip Support Library API Reference Guide*).

The CSL functions described above are present in the CSL directory. The register definitions and bit field definitions are present in the CSLR_AIF.H file.

All of the AIF register list is specified in the register overlay structure in the CSL. Each register is described as a separate structure. The MMRs are programmed by populating these structures.

8 Registers

Note: The endian configuration of the AIF is ignored by the VBUSP configuration bus interface since it only supports 32-bit accesses. Therefore, the register reads and writes by the CPU are the same irrespective of the endianness.

Antenna interface registers are classified into the following:

- Constant Registers
  - Peripheral ID Register
- Configuration Registers
  - Global Configuration Registers
  - SERDES Configuration Registers
  - Link Configuration Registers
  - Rx MAC Configuration Registers
  - Tx MAC Configuration Registers
  - Combiner/Decombiner Configuration Registers
  - Aggregator Configuration Registers
  - CPRI Input Data Format Converter Configuration Registers
  - CPRI Output Data Format Converter Configuration Registers
  - Protocol Decoder Configuration Registers
  - Protocol Encoder Configuration Registers
  - Data Buffer Configuration Registers
  - Exception Event Handler Configuration Registers
- Status Registers
  - SERDES Status Registers
  - Rx MAC Status Registers
  - Tx MAC Status Registers
  - Combiner/Decombiner Status Registers
  - Aggregator Status Registers
  - Data Buffer Status Registers
  - Exception Event Handler Status Registers
  - VBUSP Configuration Bus Interface Status Registers
  - VBUSP DMA Bus Interface Status Registers

Table 68 lists the memory map registers of Antenna Interface.

Table 68. Antenna Interface Registers

Hex Address	Acronym	Register Description	Section
02BC 0000h	AIF_PID	AIF Peripheral Identification Register	Section 8.1.1
02BC 0004h	AIF_GLOBAL_CFG	AIF Global Configuration Register	Section 8.2.1
02BC 0008h	AIF_EMU_CNTL	AIF Emulation Control Register	Section 8.2.2
02BC 7000h	AIF_SERDES0_PLL_CFG	SERDES 0 PLL Configuration Register	Section 8.2.3.1
02BC 7004h	AIF_SERDES1_PLL_CFG	SERDES 1 PLL Configuration Register	Section 8.2.3.2
02BC 7008h	AIF_SERDES0_TST_CFG	SERDES 0 Test Configuration Register	Section 8.2.3.3
02BC 700Ch	AIF_SERDES1_TST_CFG	SERDES 1 Test Configuration Register	Section 8.2.3.4
02BC 4000h	LINK0_CFG	Link 0 Configuration Register	Section 8.2.4
02BC 4800h	LINK1_CFG	Link 1 Configuration Register	Section 8.2.4
02BC 5000h	LINK2_CFG	Link 2 Configuration Register	Section 8.2.4
02BC 5800h	LINK3_CFG	Link 3 Configuration Register	Section 8.2.4
02BC 6000h	LINK4_CFG	Link 4 Configuration Register	Section 8.2.4
02BC 6800h	LINK5_CFG	Link 5 Configuration Register	Section 8.2.4

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BC 8000h	RM_LINK0_CFG	Rx MAC Link 0 Configuration Register	Section 8.2.5.1
02BC 8004h	RM_LINK0_PI_OFFSET_CFG	Rx MAC Link 0 Pi Offset Register	Section 8.2.5.2
02BC 8008h	RM_LINK0_LOS_THOLD_CFG	Rx MAC Link 0 LOS Threshold Register	Section 8.2.5.3
02BC 8800h	RM_LINK1_CFG	Rx MAC Link 1 Configuration Register	Section 8.2.5.1
02BC 8804h	RM_LINK1_PI_OFFSET_CFG	Rx MAC Link 1 Pi Offset Register	Section 8.2.5.2
02BC 8808h	RM_LINK1_LOS_THOLD_CFG	Rx MAC Link 1 LOS Threshold Register	Section 8.2.5.3
02BC 9000h	RM_LINK2_CFG	Rx MAC Link 2 Configuration Register	Section 8.2.5.1
02BC 9004h	RM_LINK2_PI_OFFSET_CFG	Rx MAC Link 2 Pi Offset Register	Section 8.2.5.2
02BC 9008h	RM_LINK2_LOS_THOLD_CFG	Rx MAC Link 2 LOS Threshold Register	Section 8.2.5.3
02BC 9800h	RM_LINK3_CFG	Rx MAC Link 3 Configuration Register	Section 8.2.5.1
02BC 9804h	RM_LINK3_PI_OFFSET_CFG	Rx MAC Link 3 Pi Offset Register	Section 8.2.5.2
02BC 9808h	RM_LINK3_LOS_THOLD_CFG	Rx MAC Link 3 LOS Threshold Register	Section 8.2.5.3
02BC A000h	RM_LINK4_CFG	Rx MAC Link 4 Configuration Register	Section 8.2.5.1
02BC A008h	RM_LINK4_LOS_THOLD_CFG	Rx MAC Link 4 LOS Threshold Register	Section 8.2.5.3
02BC A800h	RM_LINK5_CFG	Rx MAC Link 5 Configuration Register	Section 8.2.5.1
02BC A804h	RM_LINK5_PI_OFFSET_CFG	Rx MAC Link 5 Pi Offset Register	Section 8.2.5.2
02BC A808h	RM_LINK5_LOS_THOLD_CFG	Rx MAC Link 5 LOS Threshold Register	Section 8.2.5.3
02BC B000h	RM_ SYNC_CNT_CFG	Rx MAC Common Sync Counter Register	Section 8.2.5.4
02BC B004h	RM_ UNSYNC_CNT_CFG	Rx MAC Common UnSync Counter Register	Section 8.2.5.5
02BC C000h	TM_LINK0_0CFG	Tx MAC Link 0 Configuration Register 0	Section 8.2.6.1
02BC C004h	TM_LINK0_1CFG	Tx MAC Link 0 Configuration Register 1	Section 8.2.6.2
02BC C008h	TM_LINK0_2CFG	Tx MAC Link 0 Configuration Register 2	Section 8.2.6.3
02BC C800h	TM_LINK1_0CFG	Tx MAC Link 1 Configuration Register 0	Section 8.2.6.1
02BC C804h	TM_LINK1_1CFG	Tx MAC Link 1 Configuration Register 1	Section 8.2.6.2
02BC C808h	TM_LINK1_2CFG	Tx MAC Link 1 Configuration Register 2	Section 8.2.6.3
02BC D000h	TM_LINK2_0CFG	Tx MAC Link 2 Configuration Register 0	Section 8.2.6.1
02BC D004h	TM_LINK2_1CFG	Tx MAC Link 2 Configuration Register 1	Section 8.2.6.2
02BC D008h	TM_LINK2_2CFG	Tx MAC Link 2 Configuration Register 2	Section 8.2.6.3
02BC D800h	TM_LINK3_0CFG	Tx MAC Link 3 Configuration Register 0	Section 8.2.6.1
02BC D804h	TM_LINK3_1CFG	Tx MAC Link 3 Configuration Register 1	Section 8.2.6.2
02BC D808h	TM_LINK3_2CFG	Tx MAC Link 3 Configuration Register 2	Section 8.2.6.3
02BC E000h	TM_LINK4_0CFG	Tx MAC Link 4 Configuration Register 0	Section 8.2.6.1
02BC E004h	TM_LINK4_1CFG	Tx MAC Link 4 Configuration Register 1	Section 8.2.6.2
02BC E008h	TM_LINK4_2CFG	Tx MAC Link 4 Configuration Register 2	Section 8.2.6.3
02BC E800h	TM_LINK5_0CFG	Tx MAC Link 5 Configuration Register 0	Section 8.2.6.1
02BC E804h	TM_LINK5_1CFG	Tx MAC Link 5 Configuration Register 1	Section 8.2.6.2
02BC E808h	TM_LINK5_2CFG	Tx MAC Link 5 Configuration Register 2	Section 8.2.6.3
02BD 3000h	CD_OUT_MUX_SEL_CFG	CD Output Mux Source Select and Enables Configuration Register	Section 8.2.7.1
02BD 3004h	CD_CB_SRC_SEL_CFG	CD Combiner Source Select Configuration Register	Section 8.2.7.2
02BD 3008h	CD_CB_OFFSET_CFG	CD Combiner Frame Alignment Offset Configuration Register	Section 8.2.7.3
02BD 300Ch	CD_CB_VALID_WIND_CFG	CD Frame Alignment Valid Window Configuration Register	Section 8.2.7.4
02BD 3010h	CD_DC_SRC_SEL_CFG	CD Decombiner Source Select Configuration Register	Section 8.2.7.5
02BD 3014h	CD_DC_DST_SEL_CFG	CD Decombiner Destination Select Configuration Register	Section 8.2.7.6
02BD 4000h	AG_LINK0_CFG	AG Link 0 Configuration Register	Section 8.2.8
02BD 4800h	AG_LINK1_CFG	AG Link 1 Configuration Register	Section 8.2.8
02BD 5000h	AG_LINK2_CFG	AG Link 2 Configuration Register	Section 8.2.8

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BD 5800h	AG_LINK3_CFG	AG Link 3 Configuration Register	Section 8.2.8
02BD 6000h	AG_LINK4_CFG	AG Link 4 Configuration Register	Section 8.2.8
02BD 6800h	AG_LINK5_CFG	AG Link 5 Configuration Register	Section 8.2.8
02BE 7000h	PD_0_CFG	PD Configuration #0 Register	Section 8.2.11
02BE 7004h	PD_1_CFG	PD Configuration #1 Register	Section 8.2.11.2
02BE 7008h	PD_ADR_MUX_SEL_CFG	PD OBSAI Adr Mux Select Register	Section 8.2.11.3
02BE 700Ch	PD_TYPE_CIR_LUT_CFG	PD Type CirSw Capture Enable LUT Register	Section 8.2.11.4
02BE 7010h	PD_TYPE_PKT_LUT_CFG	PD Type PktSw Capture Enable LUT Register	Section 8.2.11.5
02BE 7014h	PD_TYPE_ERR_LUT_CFG	PD Type Error Register	Section 8.2.11.6
02BE 7800h to 7FFCh	PD_ADR_LUT	PD Address Look-Up Table RAM	Section 8.2.11.7
02BE 4000h	PD_LINK0_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 4004h	PD_LINK0_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 4008h	PD_LINK0_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 400Ch	PD_LINK0_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Section 8.2.11.11
02BE 4010h	PD_LINK0_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.12
02BE 4800h	PD_LINK1_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 4804h	PD_LINK1_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 4808h	PD_LINK1_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 480Ch	PD_LINK1_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Section 8.2.11.11
02BE 4810h	PD_LINK1_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.11
02BE 5000h	PD_LINK2_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 5004h	PD_LINK2_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 5008h	PD_LINK2_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 500Ch	PD_LINK2_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Figure 118
02BE 5010h	PD_LINK2_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.12
02BE 5800h	PD_LINK3_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 5804h	PD_LINK3_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 5808h	PD_LINK3_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 580Ch	PD_LINK3_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Section 8.2.11.11
02BE 5810h	PD_LINK3_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.12
02BE 6000h	PD_LINK4_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 6004h	PD_LINK4_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 6008h	PD_LINK4_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 600Ch	PD_LINK4_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Section 8.2.11.11
02BE 6010h	PD_LINK4_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.12
02BE 6800h	PD_LINK5_84CNT_LUT0_CFG	PD 84 Count Look-Up Table bits [31:0]	Section 8.2.11.8
02BE 6804h	PD_LINK5_84CNT_LUT1_CFG	PD 84 Count Look-Up Table bits [63:32]	Section 8.2.11.9
02BE 6808h	PD_LINK5_84CNT_LUT2_CFG	PD 84 Count Look-Up Table bits [83:64]	Section 8.2.11.10
02BE 680Ch	PD_LINK5_CPRI_SI_LUT0_CFG	PD CPRI Stream Index LUT0 Register	Section 8.2.11.11
02BE 6810h	PD_LINK5_CPRI_SI_LUT1_CFG	PD CPRI Stream Index LUT1 Register	Section 8.2.11.12
02BE B000h	PE_CFG	PE Configuration Register	Section 8.2.12.1
02BE 8000h	PE_LINK0_84_EN_LUT0_CFG	PE 84 Count Message Enable bits [31:0]	Section 8.2.12.2
02BE 8004h	PE_LINK0_84_EN_LUT1_CFG	PE 84 Count Message Enable bits [63:32]	Section 8.2.12.3
02BE 8008h	PE_LINK0_84_EN_LUT2_CFG	PE 84 Count Message Enable bits [83:64]	Section 8.2.12.4
02BE 800Ch	PE_LINK0_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE 8010h	PE_LINK0_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BE 8200h to 834Ch	PE_LINK0_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE 8400h to 8450h	PE_LINK0_ID_LUT0	PE Identity LUT Part 0 RAM	Section 8.2.12.8
02BE 8500h to 8550h	PE_LINK0_ID_LUT1	PE Identity LUT Part 1 RAM	Section 8.2.12.9
02BE 8800h	PE_LINK1_84_EN_LUT0_CFG	PE 84 Count Message Enable bits [31:0]	Section 8.2.12.2
02BE 8804h	PE_LINK1_84_EN_LUT1_CFG	PE 84 Count Message Enable bits [63:32]	Section 8.2.12.3
02BE 8808h	PE_LINK1_84_EN_LUT2_CFG	PE 84 Count Message Enable bits [83:64]	Section 8.2.12.4
02BE 880Ch	PE_LINK1_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE 8810h	PE_LINK1_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6
02BE 8A00h to 8B4Ch	PE_LINK1_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE 8C00h to 8C50h	PE_LINK1_ID_LUT0	PE Identity LUT Part 0 RAM	Section 8.2.12.8
02BE 8D00h to 8D50h	PE_LINK1_ID_LUT1	PE Identity LUT Part 1 RAM	Section 8.2.12.9
02BE 9000h	PE_LINK2_84_EN_LUT0_CFG	PE 84 Count message Enable bits [31: 0]	Section 8.2.12.2
02BE 9004h	PE_LINK2_84_EN_LUT1_CFG	PE 84 Count message Enable bits [63 : 32]	Section 8.2.12.3
02BE 9008h	PE_LINK2_84_EN_LUT2_CFG	PE 84 Count message Enable bits [83 : 64]	Section 8.2.12.4
02BE 900Ch	PE_LINK2_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE 9010h	PE_LINK2_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6
02BE 9200h to 934Ch	PE_LINK2_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE 9400h to 9450h	PE_LINK2_ID_LUT0	PE Identity LUT part 0 RAM	Section 8.2.12.8
02BE 9500h to 9550h	PE_LINK2_ID_LUT1	PE Identity LUT part 1 RAM	Section 8.2.12.9
02BE 9800h	PE_LINK3_84_EN_LUT0_CFG	PE 84 Count message Enable bits [31:0]	Section 8.2.12.2
02BE 9804h	PE_LINK3_84_EN_LUT1_CFG	PE 84 Count message Enable bits [63:32]	Section 8.2.12.3
02BE 9808h	PE_LINK3_84_EN_LUT2_CFG	PE 84 Count message Enable bits [83:64]	Section 8.2.12.4
02BE 980Ch	PE_LINK3_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE 9810h	PE_LINK3_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6
02BE 9A00h to 9B4Ch	PE_LINK3_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE 9C00h to 9C50h	PE_LINK3_ID_LUT0	PE Identity LUT Part 0 RAM	Section 8.2.12.8
02BE 9D00h to 9D50h	PE_LINK3_ID_LUT1	PE Identity LUT Part 1 RAM	Section 8.2.12.9
02BE A000h	PE_LINK4_84_EN_LUT0_CFG	PE 84 Count Message Enable bits [31:0]	Section 8.2.12.2
02BE A004h	PE_LINK4_84_EN_LUT1_CFG	PE 84 Count Message Enable bits [63:32]	Section 8.2.12.3
02BE A008h	PE_LINK4_84_EN_LUT2_CFG	PE 84 Count Message Enable bits [83:64]	Section 8.2.12.4
02BE A00Ch	PE_LINK4_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE A010h	PE_LINK4_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6
02BE A200h to A34Ch	PE_LINK4_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE A400h to A450h	PE_LINK4_ID_LUT0	PE Identity LUT Part 0 RAM	Section 8.2.12.8
02BE A500h to A550h	PE_LINK4_ID_LUT1	PE Identity LUT Part 1 RAM	Section 8.2.12.9
02BE A800h	PE_LINK5_84_EN_LUT0_CFG	PE 84 Count Message Enable bits [31:0]	Section 8.2.12.2
02BE A804h	PE_LINK5_84_EN_LUT1_CFG	PE 84 Count Message Enable bits [63:32]	Section 8.2.12.3

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BE A808h	PE_LINK5_84_EN_LUT2_CFG	PE 84 Count Message Enable bits [83:64]	Section 8.2.12.4
02BE A80Ch	PE_LINK5_TERM_CNT01_CFG	PE Transmission Rule Terminal Count 0 & 1	Section 8.2.12.5
02BE A810h	PE_LINK5_TERM_CNT23_CFG	PE Transmission Rule Terminal Count 2 & 3	Section 8.2.12.6
02BE AA00h to AB4Ch	PE_LINK5_84CNT_LUT	PE 84 Count LUT RAM	Section 8.2.12.7
02BE AC00h to AC50h	PE_LINK5_ID_LUT0	PE Identity LUT Part 0 RAM	Section 8.2.12.8
02BE AD00h to AD50h	PE_LINK5_ID_LUT1	PE Identity LUT Part 1 RAM	Section 8.2.12.9
02BE 3004h	DB_GENERIC_CFG	DB Configuration Register	Section 8.2.13.1
02BE 3008h	DB_DMA_QUE_CLR_CFG	DB DMA Depth Clear Register	Section 8.2.13.2
02BE 300Ch	DB_DMA_CNT_CLR_CFG	DB DMA Depth Clear Register	Section 8.2.13.3
02BE 3010h	DB_OUT_PKTSW_EN_CFG	DB Outbound Packet-Switched FIFO Enable Register	Section 8.2.13.4
02BE 3014h	DB_OUT_PKTSW_FLUSH_CFG	DB Inbound Packet-Switched FIFO Flush Register	Section 8.2.13.5
02BE 3018h	DB_IN_FIFO_EVNT_CFG	DB Inbound Packet-Switched FIFO Empty_n Event Depth	Section 8.2.13.6
02BE 301Ch	DB_IN_FIFO_SIZE_CFG	DB Inbound Packet-Switched FIFO Depth Register	Section 8.2.13.7
02BE 3020h	DB_FORCE_SYSEVENT_CFG	DB Force System Events Register	Section 8.2.13.8
02BE 3024h	DB_OUTB_TRK_AUTOSYNC_CFG	DB Tracker PE Auto Sync Control Register	Section 8.2.13.9
02BE 3028h	DB_INB_TRK_AUTOSYNC_CFG	DB Tracker PD Auto Sync Control Register	Section 8.2.13.10
02BD 8000h	CI_LINK0_CFG	CI Link 0 Configuration Register	Section 8.2.9
02BD 8800h	CI_LINK1_CFG	CI Link 1 Configuration Register	Section 8.2.9
02BD 9000h	CI_LINK2_CFG	CI Link 2 Configuration Register	Section 8.2.9
02BD 9800h	CI_LINK3_CFG	CI Link 3 Configuration Register	Section 8.2.9
02BD A000h	CI_LINK4_CFG	CI Link 4 Configuration Register	Section 8.2.9
02BD A800h	CI_LINK5_CFG	CI Link 5 Configuration Register	Section 8.2.9
02BD C000h	CO_LINK0_CFG	CO Link 0 Configuration Register	Section 8.2.10
02BD C800h	CO_LINK1_CFG	CO Link 1 Configuration Register	Section 8.2.10
02BD D000h	CO_LINK2_CFG	CO Link 2 Configuration Register	Section 8.2.10
02BD D800h	CO_LINK3_CFG	CO Link 3 Configuration Register	Section 8.2.10
02BD E000h	CO_LINK4_CFG	CO Link 4 Configuration Register	Section 8.2.10
02BD E800h	CO_LINK5_CFG	CO Link 5 Configuration Register	Section 8.2.10
02BF 3000h	EE_CFG	EE Configuration Register	Section 8.2.14.1
02BF 3004h	EE_LINK_SEL_A_EV2	EE AI_EVENT[2] Link Select Register A	Section 8.2.14.2
02BF 3008h	EE_LINK_SEL_B_EV2	EE AI_EVENT[2] Link Select Register B	Section 8.2.14.3
02BF 300Ch	EE_LINK_SEL_A_EV3	EE AI_EVENT[3] Link Select Register A	Section 8.2.14.2
02BF 3010h	EE_LINK_SEL_B_EV3	EE AI_EVENT[3] Link Select Register B	Section 8.2.14.3
02BF 3014h	EE_INT_END	EE End of Interrupt Register	Section 8.2.14.4
		Status Registers	
02BC 7080h	SERDES_STS	SERDES Status Register	Section 8.3.1
02BC 8080h	RM_LINK0_STSA	Rx MAC Link 0 Status Register 0	Section 8.3.2.1
02BC 8084h	RM_LINK0_STSB	Rx MAC Link 0 Status Register 1	Section 8.3.2.2
02BC 8088h	RM_LINK0_STSC	Rx MAC Link 0 Status Register 2	Section 8.3.2.3
02BC 808Ch	RM_LINK0_STSD	Rx MAC Link 0 Status Register 3	Section 8.3.2.4
02BC 8880h	RM_LINK1_STSA	Rx MAC Link 1 Status Register 0	Section 8.3.2.1
02BC 8884h	RM_LINK1_STSB	Rx MAC Link 1 Status Register 1	Section 8.3.2.2
02BC 8888h	RM_LINK1_STSC	Rx MAC Link 1 Status Register 2	Section 8.3.2.3
02BC 888Ch	RM_LINK1_STSD	Rx MAC Link 1 Status Register 3	Section 8.3.2.4
02BC 9080h	RM_LINK2_STSA	Rx MAC Link 2 Status Register 0	Section 8.3.2.1

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BC 9084h	RM_LINK2_STSB	Rx MAC Link 2 Status Register 1	Section 8.3.2.2
02BC 9088h	RM_LINK2_STSC	Rx MAC Link 2 Status Register 2	Section 8.3.2.3
02BC 908Ch	RM_LINK2_STSD	Rx MAC Link 2 Status Register 3	Section 8.3.2.4
02BC 9880h	RM_LINK3_STSA	Rx MAC Link 3 Status Register 0	Section 8.3.2.1
02BC 9884h	RM_LINK3_STSB	Rx MAC Link 3 Status Register 1	Section 8.3.2.2
02BC 9888h	RM_LINK3_STSC	Rx MAC Link 3 Status Register 2	Section 8.3.2.3
02BC 988Ch	RM_LINK3_STSD	Rx MAC Link 3 Status Register 3	Section 8.3.2.4
02BC A080h	RM_LINK4_STSA	Rx MAC Link 4 Status Register 0	Section 8.3.2.1
02BC A084h	RM_LINK4_STSB	Rx MAC Link 4 Status Register 1	Section 8.3.2.2
02BC A088h	RM_LINK4_STSC	Rx MAC Link 4 Status Register 2	Section 8.3.2.3
02BC A08Ch	RM_LINK4_STSD	Rx MAC Link 4 Status Register 3	Section 8.3.2.4
02BC A880h	RM_LINK5_STSA	Rx MAC Link 5 Status Register 0	Section 8.3.2.1
02BC A884h	RM_LINK5_STSB	Rx MAC Link 5 Status Register 1	Section 8.3.2.2
02BC A888h	RM_LINK5_STSC	Rx MAC Link 5 Status Register 2	Section 8.3.2.3
02BC A88Ch	RM_LINK5_STSD	Rx MAC Link 5 Status Register 3	Section 8.3.2.4
02BC C080h	TM_LINK0_STS	Tx MAC Link 0 Status Register	Section 8.3.3
02BC C880h	TM_LINK1_STS	Tx MAC Link 1 Status Register	Section 8.3.3
02BC D080h	TM_LINK2_STS	Tx MAC Link 2 Status Register	Section 8.3.3
02BC D880h	TM_LINK3_STS	Tx MAC Link 3 Status Register	Section 8.3.3
02BC E080h	TM_LINK4_STS	Tx MAC Link 4 Status Register	Section 8.3.3
02BC E880h	TM_LINK5_STS	Tx MAC Link 5 Status Register	Section 8.3.3
02BD 3080h	CD_STS	Combiner / Decombiner Status Register	Section 8.3.4
02BD 4004h	AG_LINK0_STS	AG Link 0 Status Register	Section 8.3.5.1
02BD 4008h	AG_LINK0_HDR_ERR_STSA	AG Link 0 Header Error Status Register 0	Section 8.3.5.2
02BD 400Ch	AG_LINK0_HDR_ERR_STSB	AG Link 0 Header Error Status Register 1	Section 8.3.5.3
02BD 4010h	AG_LINK0_HDR_ERR_STSC	AG Link 0 Header Error Status Register 2	Section 8.3.5.4
02BD 4014h	AG_LINK0_HDR_ERR_STSD	AG Link 0 Header Error Status Register 3	Section 8.3.5.5
02BD 4804h	AG_LINK1_STS	AG Link 1 Status Register	Section 8.3.5.1
02BD 4808h	AG_LINK1_HDR_ERR_STSA	AG Link 1 Header Error Status Register 0	Section 8.3.5.2
02BD 480Ch	AG_LINK1_HDR_ERR_STSB	AG Link 1 Header Error Status Register 1	Section 8.3.5.3
02BD 4810h	AG_LINK1_HDR_ERR_STSC	AG Link 1 Header Error Status Register 2	Section 8.3.5.4
02BD 4814h	AG_LINK1_HDR_ERR_STSD	AG Link 1 Header Error Status Register 3	Section 8.3.5.5
02BD 5004h	AG_LINK2_STS	AG Link 2 Status Register	Section 8.3.5.1
02BD 5008h	AG_LINK2_HDR_ERR_STSA	AG Link 2 Header Error Status Register 0	Section 8.3.5.2
02BD 500Ch	AG_LINK2_HDR_ERR_STSB	AG Link 2 Header Error Status Register 1	Section 8.3.5.3
02BD 5010h	AG_LINK2_HDR_ERR_STSC	AG Link 2 Header Error Status Register 2	Section 8.3.5.4
02BD 5014h	AG_LINK2_HDR_ERR_STSD	AG Link 2 Header Error Status Register 3	Section 8.3.5.5
02BD 5804h	AG_LINK3_STS	AG Link 3 Status Register	Section 8.3.5.1
02BD 5808h	AG_LINK3_HDR_ERR_STSA	AG Link 3 Header Error Status Register 0	Section 8.3.5.2
02BD 580Ch	AG_LINK3_HDR_ERR_STSB	AG Link 3 Header Error Status Register 1	Section 8.3.5.3
02BD 5810h	AG_LINK3_HDR_ERR_STSC	AG Link 3 Header Error Status Register 2	Section 8.3.5.4
02BD 5814h	AG_LINK3_HDR_ERR_STSD	AG Link 3 Header Error Status Register 3	Section 8.3.5.5
02BD 6004h	AG_LINK4_STS	AG Link 4 Status Register	Section 8.3.5.1
02BD 6008h	AG_LINK4_HDR_ERR_STSA	AG Link 4 Header Error Status Register 0	Section 8.3.5.2
02BD 600Ch	AG_LINK4_HDR_ERR_STSB	AG Link 4 Header Error Status Register 1	Section 8.3.5.3
02BD 6010h	AG_LINK4_HDR_ERR_STSC	AG Link 4 Header Error Status Register 2	Section 8.3.5.4
02BD 6014h	AG_LINK4_HDR_ERR_STSD	AG Link 4 Header Error Status Register 3	Section 8.3.5.5

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BD 6804h	AG_LINK5_STS	AG Link 5 Status Register	Section 8.3.5.1
02BD 6808h	AG_LINK5_HDR_ERR_STSA	AG Link 5 Header Error Status Register 0	Section 8.3.5.2
02BD 680Ch	AG_LINK5_HDR_ERR_STSB	AG Link 5 Header Error Status Register 1	Section 8.3.5.3
02BD 6810h	AG_LINK5_HDR_ERR_STSC	AG Link 5 Header Error Status Register 2	Section 8.3.5.4
02BD 6814h	AG_LINK5_HDR_ERR_STSD	AG Link 5 Header Error Status Register 3	Section 8.3.5.5
02BE 3040h	DB_IN_DMA_CNT0_STS	DB Inbound DMA Count 0 Register	Section 8.3.6.1
02BE 3044h	DB_IN_DMA_CNT1_STS	DB Inbound DMA Count 1 Register	Section 8.3.6.2
02BE 3048h	DB_IN_DMA_CNT2_STS	DB Inbound DMA Count 2 Register	Section 8.3.6.3
02BE 304Ch	DB_OUT_DMA_CNT0_STS	DB Outbound DMA Count 0 Register	Section 8.3.6.4
02BE 3050h	DB_OUT_DMA_CNT1_STS	DB Outbound DMA Count 1 Register	Section 8.3.6.5
02BE 3054h	DB_OUT_DMA_CNT2_STS	DB Outbound DMA Count 2 Register	Section 8.3.6.6
02BE 3058h	DB_IN_DMA_DEPTH_STS	DB Inbound DMA Burst Available Register	Section 8.3.6.7
02BE 305Ch	DB_OUT_DMA_DEPTH_STS	DB Outbound DMA Burst Available Register	Section 8.3.6.8
02BE 3060h	DB_OUT_PKTSW_STS	DB Outbound Packet-Switched FIFO status Register	Section 8.3.6.9
02BE 3064h	DB_OUT_PKTSW_DEPTH_STS	Data Buffer Outbound Packet-Switched FIFO Depth Register	Section 8.3.6.10
02BE 3068h	DB_OUT_PKTSW_NE_STS	Data Buffer Outbound Packet-Switched FIFO Not Empty Register	Section 8.3.6.11
02BE 306Ch	DB_DATA_TRACE_STS	Data Buffer Trace Buffer Status Register	Section 8.3.6.12
02BE 3080h	DB_OUT_PKTSW_HEAD0_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 3084h	DB_OUT_PKTSW_HEAD1_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 3088h	DB_OUT_PKTSW_HEAD2_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 308Ch	DB_OUT_PKTSW_HEAD3_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 3090h	DB_OUT_PKTSW_HEAD4_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 3094h	DB_OUT_PKTSW_HEAD5_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 3098h	DB_OUT_PKTSW_HEAD6_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 309Ch	DB_OUT_PKTSW_HEAD7_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30A0h	DB_OUT_PKTSW_HEAD8_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30A4h	DB_OUT_PKTSW_HEAD9_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30A8h	DB_OUT_PKTSW_HEAD10_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30ACh	DB_OUT_PKTSW_HEAD11_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30B0h	DB_OUT_PKTSW_HEAD12_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30B4h	DB_OUT_PKTSW_HEAD13_STS	DB Outbound Packet-Switched FIFO Head Pointers Registers	Section 8.3.6.13
02BE 30B8h	DB_OUT_PKTSW_HEAD14_STS	DB Outbound Packet-Switched FIFO Head Ptrs Registers	Section 8.3.6.13
02BE 30C0h	DB_OUT_PKTSW_TAIL0_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30C4h	DB_OUT_PKTSW_TAIL1_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30C8h	DB_OUT_PKTSW_TAIL2_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30CCh	DB_OUT_PKTSW_TAIL3_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BE 30D0h	DB_OUT_PKTSW_TAIL4_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30D4h	DB_OUT_PKTSW_TAIL5_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30D8h	DB_OUT_PKTSW_TAIL6_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30DCh	DB_OUT_PKTSW_TAIL7_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30E0h	DB_OUT_PKTSW_TAIL8_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30E4h	DB_OUT_PKTSW_TAIL9_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30E8h	DB_OUT_PKTSW_TAIL10_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30ECh	DB_OUT_PKTSW_TAIL11_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02B 30F0h	DB_OUT_PKTSW_TAIL12_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30F4h	DB_OUT_PKTSW_TAIL13_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BE 30F8h	DB_OUT_PKTSW_TAIL14_STS	DB Outbound Packet-Switched FIFO Tail Ptrs Registers	Section 8.3.6.14
02BF 3080h	EE_AI_RUN	EE AIF Run Register	Section 8.3.7.1
02BF 0000h	EE_LINK0_IRS_A	EE Link 0 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 0004h	EE_LINK0_IRS_B	EE Link 0 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 0008h	EE_LINK0_IMS_A_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 000Ch	EE_LINK0_IMS_B_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 0010h	EE_LINK0_IMS_A_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 0014h	EE_LINK0_IMS_B_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 0018h	EE_LINK0_MSK_SET_A_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 001Ch	EE_LINK0_MSK_SET_B_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 0020h	EE_LINK0_MSK_SET_A_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 0024h	EE_LINK0_MSK_SET_B_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 0028h	EE_LINK0_MSK_CLR_A_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 002Ch	EE_LINK0_MSK_CLR_B_EV0	EE Link 0 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 0030h	EE_LINK0_MSK_CLR_A_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 0034h	EE_LINK0_MSK_CLR_B_EV1	EE Link 0 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 0800h	EE_LINK1_IRS_A	EE Link 1 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 0804h	EE_LINK1_IRS_B	EE Link 1 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 0808h	EE_LINK1_IMS_A_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 080Ch	EE_LINK1_IMS_B_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 0810h	EE_LINK1_IMS_A_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 0814h	EE_LINK1_IMS_B_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 0818h	EE_LINK1_MSK_SET_A_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 081Ch	EE_LINK1_MSK_SET_B_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BF 0820h	EE_LINK1_MSK_SET_A_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 0824h	EE_LINK1_MSK_SET_B_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 0828h	EE_LINK1_MSK_CLR_A_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 082Ch	EE_LINK1_MSK_CLR_B_EV0	EE Link 1 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 0830h	EE_LINK1_MSK_CLR_A_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 0834h	EE_LINK1_MSK_CLR_B_EV1	EE Link 1 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 1000h	EE_LINK2_IRS_A	EE Link 2 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 1004h	EE_LINK2_IRS_B	EE Link 2 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 1008h	EE_LINK2_IMS_A_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 100Ch	EE_LINK2_IMS_B_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 1010h	EE_LINK2_IMS_A_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 1014h	EE_LINK2_IMS_B_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 1018h	EE_LINK2_MSK_SET_A_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 101Ch	EE_LINK2_MSK_SET_B_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 1020h	EE_LINK2_MSK_SET_A_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 1024h	EE_LINK2_MSK_SET_B_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 1028h	EE_LINK2_MSK_CLR_A_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 102Ch	EE_LINK2_MSK_CLR_B_EV0	EE Link 2 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 1030h	EE_LINK2_MSK_CLR_A_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 1034h	EE_LINK2_MSK_CLR_B_EV1	EE Link 2 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 1800h	EE_LINK3_IRS_A	EE Link 3 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 1804h	EE_LINK3_IRS_B	EE Link 3 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 1808h	EE_LINK3_IMS_A_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 180Ch	EE_LINK3_IMS_B_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 1810h	EE_LINK3_IMS_A_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 1814h	EE_LINK3_IMS_B_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 1818h	EE_LINK3_MSK_SET_A_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 181Ch	EE_LINK3_MSK_SET_B_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 1820h	EE_LINK3_MSK_SET_A_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 1824h	EE_LINK3_MSK_SET_B_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BF 1828h	EE_LINK3_MSK_CLR_A_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 182Ch	EE_LINK3_MSK_CLR_B_EV0	EE Link 3 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 1830h	EE_LINK3_MSK_CLR_A_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 1834h	EE_LINK3_MSK_CLR_B_EV1	EE Link 3 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 2000h	EE_LINK4_IRS_A	EE Link 4 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 2004h	EE_LINK4_IRS_B	EE Link 4 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 2008h	EE_LINK4_IMS_A_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 200Ch	EE_LINK4_IMS_B_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 2010h	EE_LINK4_IMS_A_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 2014h	EE_LINK4_IMS_B_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Masked Status Register B	**Section 8.3.7.6**
02BF 2018h	EE_LINK4_MSK_SET_A_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 201Ch	EE_LINK4_MSK_SET_B_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 2020h	EE_LINK4_MSK_SET_A_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 2024h	EE_LINK4_MSK_SET_B_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 2028h	EE_LINK4_MSK_CLR_A_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 202Ch	EE_LINK4_MSK_CLR_B_EV0	EE Link 4 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 2030h	EE_LINK4_MSK_CLR_A_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 2034h	EE_LINK4_MSK_CLR_B_EV1	EE Link 4 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 2800h	EE_LINK5_IRS_A	EE Link 5 Interrupt Source Raw Status Register A	Section 8.3.7.2
02BF 2804h	EE_LINK5_IRS_B	EE Link 5 Interrupt Source Raw Status Register B	Section 8.3.7.3
02BF 2808h	EE_LINK5_IMS_A_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 280Ch	EE_LINK5_IMS_B_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 2810h	EE_LINK5_IMS_A_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 2814h	EE_LINK5_IMS_B_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 2818h	EE_LINK5_MSK_SET_A_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 281Ch	EE_LINK5_MSK_SET_B_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 2820h	EE_LINK5_MSK_SET_A_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 2824h	EE_LINK5_MSK_SET_B_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 2828h	EE_LINK5_MSK_CLR_A_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 282Ch	EE_LINK5_MSK_CLR_B_EV0	EE Link 5 AI_EVENT[0] Interrupt Source Mask Clear Register B	Section 8.3.7.12

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BF 2830h	EE_LINK5_MSK_CLR_A_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 2834h	EE_LINK5_MSK_CLR_B_EV1	EE Link 5 AI_EVENT[1] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 3100h	EE_COMMON_IRS	EE Common Interrupt Source Raw Status Register	Section 8.3.7.4
02BF 3104h	EE_COMMON_IMS_EV0	EE Common AI_EVENT[0] Interrupt Source Masked Status Register	Section 8.3.7.7
02BF 3108h	EE_COMMON_IMS_EV1	EE Common AI_EVENT[1] Interrupt Source Masked Status Register	Section 8.3.7.7
02BF 310Ch	EE_LINK_IMS_A_EV2	EE Per Link AI_EVENT[2] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 3110h	EE_LINK_IMS_B_EV2	EE Per Link AI_EVENT[2] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 3114h	EE_COMMON_IMS_EV2	EE Common AI_EVENT[2] Interrupt Source Masked Status Register	Section 8.3.7.7
02BF 3118h	EE_LINK_IMS_A_EV3	EE Per Link AI_EVENT[3] Interrupt Source Masked Status Register A	Section 8.3.7.5
02BF 311Ch	EE_LINK_IMS_B_EV3	EE Per Link AI_EVENT[3] Interrupt Source Masked Status Register B	Section 8.3.7.6
02BF 3120h	EE_COMMON_IMS_EV3	EE Common AI_EVENT[3] Interrupt Source Masked Status Register	Section 8.3.7.7
02BF 3124h	EE_COMMON_MSK_SET_EV0	EE Common AI_EVENT[0] Interrupt Source Mask Set Register	Section 8.3.7.10
02BF 3128h	EE_COMMON_MSK_SET_EV1	EE Common AI_EVENT[1] Interrupt Source Mask Set Register	Section 8.3.7.10
02BF 312Ch	EE_LINK_MSK_SET_A_EV2	EE Per Link AI_EVENT[2] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 3130h	EE_LINK_MSK_SET_B_EV2	EE Per Link AI_EVENT[2] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 3134h	EE_COMMON_MSK_SET_EV2	EE Common AI_EVENT[2] Interrupt Source Mask Set Register	Section 8.3.7.10
02BF 3138h	EE_LINK_MSK_SET_A_EV3	EE Per Link AI_EVENT[3] Interrupt Source Mask Set Register A	Section 8.3.7.8
02BF 313Ch	EE_LINK_MSK_SET_B_EV3	EE Per Link AI_EVENT[3] Interrupt Source Mask Set Register B	Section 8.3.7.9
02BF 3140h	EE_COMMON_MSK_SET_EV3	EE Common AI_EVENT[3] Interrupt Source Mask Set Register	Section 8.3.7.10
02BF 3144h	EE_COMMON_MSK_CLR_EV0	EE Common AI_EVENT[0] Interrupt Source Mask Clear Register	Section 8.3.7.13
02BF 3148h	EE_COMMON_MSK_CLR_EV1	EE Common AI_EVENT[1] Interrupt Source Mask Clear Register	Section 8.3.7.13
02BF 314Ch	EE_LINK_MSK_CLR_A_EV2	EE Per Link AI_EVENT[2] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 3150h	EE_LINK_MSK_CLR_B_EV2	EE Per Link AI_EVENT[2] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 3154h	EE_COMMON_MSK_CLR_EV2	EE Common AI_EVENT[2] Interrupt Source Mask Clear Register	Section 8.3.7.10
02BF 3158h	EE_LINK_MSK_CLR_A_EV3	EE Per Link AI_EVENT[3] Interrupt Source Mask Clear Register A	Section 8.3.7.11
02BF 315Ch	EE_LINK_MSK_CLR_B_EV3	EE Per Link AI_EVENT[3] Interrupt Source Mask Clear Register B	Section 8.3.7.12
02BF 3160h	EE_COMMON_MSK_CLR_EV3	EE Common AI_EVENT[3] Interrupt Source Mask Clear Register	Section 8.3.7.10
02BF 3200h	EE_INT_VECT_EV0	EE AI_EVENT[0] Interrupt Vector Register	Section 8.3.7.14
02BF 3204h	EE_INT_VECT_EV1	EE AI_EVENT[1] Interrupt Vector Register	Section 8.3.7.14
02BF 3208h	EE_INT_VECT_EV2	EE AI_EVENT[2] Interrupt Vector Register	Section 8.3.7.15

Table 68. Antenna Interface Registers (continued)

Hex Address	Acronym	Register Description	Section
02BF 320Ch	EE_INT_VECT_EV3	EE AI_EVENT[3] Interrupt Vector Register	Section 8.3.7.15
02BC 000Ch	VC_BUS_ERR	VC Bus Error Register	Section 8.3.8.1
02BE C000h	VD_RD_BUSERR	VD Read-Only Bus Error Register	Section 8.3.9.1
02BE C004h	VD_WR_BUSERR	VD WRite-Only Bus Error Register	Section 8.3.9.2

8.1 Constant Registers

These registers contain constant value fields.

- These registers are read only. Typically the field values indicate revision status or mod-enabled status.
- The field values can either be hard-wired like the peripheral ID or be set through module tie-offs.

8.1.1 AIF Peripheral ID Register (AIF_PID)

The peripheral identification register (PID) is a constant register that contains the ID and ID revision number of the AIF. All bits within this register are read-only.

The AIF peripheral ID register (AIF_PID) is shown in Figure 83 and described in Table 69.

Figure 83. AIF Peripheral ID Register (AIF_PID)

31–30	29–28	27–16
Scheme	Reserved	Function
R-x	R-x	R-x

15–11	10–8	7–6	5–0
RTL	Major	Custom	Minor
R-x	R-x	R-x	R-x

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 69. AIF Peripheral ID Register (AIF_PID) Field Descriptions

Bit	Field	Value	Description
31-30	Scheme	1	Hard coded to “01” to indicate the current scheme.
29-28	Reserved	0	Reserved
27-16	Function	0x800	Function code assigned to AIF. Hard coded to 0x800.
15-11	RTL	0x200	RTL Version (R) code
10-8	Major	0	Major revision (X) code
7-6	Custom	0	Custom version code
5-0	Minor	0xC	Minor revision (Y) code

8.2 Configuration Registers

The following are the types of configuration registers:

8.2.1 AIF Global Configuration Registers (AIF_GLOBAL_CFG)

The AIF global configuration register (AIF_GLOBAL_CFG) is shown in Figure 84 and described in Table 70.

Figure 84. AIF Global Configuration Register (AIF_GLOBAL_CFG)

31	16
Reserved	
R-0	

15	9	8	7	6	4	3	1	0
Reserved				RM_RAW_DATA_SEL		Reserved		OP_MODE
R-0				R/W-0		R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 70. AIF Global Configuration Register (AIF_GLOBAL_CFG) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	RM_RAW_DATA_SEL	0-7h	Select one of six links of raw (not gated) Rx MAC output data used for the data tracing feature.
3-1	Reserved	0	Reserved
0	OP_MODE		Select Operating mode. This bit must be set at device configuration and not changed (affects Rx MAC, Tx MAC, AG, CD, CO, CI, PD, PE, and DB).
		0	OBSAI
		1	CPRI

8.2.2 AIF Emulation Control Register (AIF_EMU_CNTL)

The AIF emulation control register (AIF_EMU_CNTL) is shown in Figure 85 and described in Table 71.

Figure 85. AIF Emulation Control Register (AIF_EMU_CNTL)

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		RT_SEL	Reserved	FREERUN
R-0		R/W-0	R-0	R/W-1

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 71. AIF Emulation Control Register (AIF_EMU_CNTL) Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	RT_SEL		Source of emulation suspend to AIF
		0	AIF_EMUSUSP (ignore AIF_EMUSUSP_RT)
		1	AIF_EMUSUSP_RT (ignore AIF_EMUSUSP)
1	Reserved	0	Reserved
0	FREERUN		Controls whether or not the AIF responds to the emulation suspend signal selected by RT_SEL below
		0	AIF responds to emulation suspend signal selected by RT_SEL
		1	AIF ignores emulation suspend signal selected by RT_SEL

8.2.3 SERDES Configuration Registers

8.2.3.1 SERDES 0 PLL Configuration Register (AIF_SERDES0_PLL_CFG)

The SERDES 0 PLL configuration register (AIF_SERDES0_PLL_CFG) is shown in Figure 86 and described in Table 72.

Figure 86. SERDES 0 PLL Configuration Register (AIF_SERDES0_PLL_CFG)

31	16
Reserved	
R-0	

15	10	9	8	7	5	4	1	0
Reserved		LOOP_BW		Reserved		MPY_FACTOR		PLL_EN
R-0		R/W-0		R-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 72. SERDES 0 PLL Configuration Register (AIF_SERDES0_PLL_CFG) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-8	LOOP_BW	0-3h	Specify loop bandwidth settings
7-5	Reserved	0	Reserved
4-1	MPY_FACTOR	0-Fh	Select PLL multiply factor (AIF supports between 4 and 25). See Table 24.
0	PLL_EN		Enable PLL

8.2.3.2 SERDES 1 PLL Configuration Register (AIF_SERDES1_PLL_CFG)

The SERDES 1 PLL configuration register (AIF_SERDES1_PLL_CFG) is shown in Figure 87 and described in Table 73.

Figure 87. SERDES 1 PLL Configuration Register (AIF_SERDES1_PLL_CFG)

31	16
Reserved	
R-0	

15	10	9	8	7	5	4	1	0
Reserved		LOOP_BW		Reserved		MPY_FACTOR		PLL_EN
R-0		R/W-0		R-0		R/W-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 73. SERDES 1 PLL Configuration Register (AIF_SERDES1_PLL_CFG) Field Descriptions

Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-8	LOOP_BW	0-3h	Specify loop bandwidth settings
7-5	Reserved	0	Reserved
4-1	MPY_FACTOR	0-Fh	Select PLL multiply factor (AIF supports between 4 and 25). See Table 24.
0	PLL_EN		Enable PLL

8.2.3.3 SERDES 0 Test Configuration Register (AIF_SERDES0_TST_CFG)

The SERDES 0 test configuration register (AIF_SERDES0_TST_CFG) is shown in Figure 88 and described in Table 74.

Figure 88. SERDES 0 Test Configuration Register (AIF_SERDES0_TST_CFG)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved	INVPATT	Rate		Reserved			
R-0	R/W-0	R/W-0		R-0			

7	6	5	4	3	2	1	0
Loopback		CLK_BYP		EN_RX_PATT	EN_TX_PATT	TEST_PATT	
R/W-0		R/W-0		R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 74. SERDES 0 Test Configuration Register (AIF_SERDES0_TST_CFG) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	INVPATT		Polarity of pattern generator
		0	Does not invert polarity of pattern generator
		1	Inverts polarity of pattern generator
13-12	Rate	0-3h	Selects rate for TX pattern generator
11-8	Reserved	0	Reserved
7-6	Loopback		Selects internal or bump/pad loopback
		0h	Disabled
		1h	Bump/Pad loopback
		2h	Inner loopback, CML driver disabled
		3h	Inner loopback, CML driver enabled
5-4	CLK_BYP		Facilitates bypassing of the PLL clock and bypassing of the recovered receive clock
		0h	No bypass
		1h	Reserved
		2h	Functional bypass
		3h	REFCLK Observe. The PLL is bypassed by REFCLKP/N
3	EN_RX_PATT		Generation of test patterns in the receiver
		0	Disables generation of test patterns in the receiver
		1	Enables generation of test patterns in the receiver
2	EN_TX_PATT		Generation of test patterns in the transmitter
		0	Disables generation of test patterns in the transmitter
		1	Enables generation of test patterns in the transmitter
1-0	TEST_PATT	0-3h	Test Pattern selects one of two PRBS or two clock test patterns

8.2.3.4 SERDES 1 Test Configuration Register (AIF_SERDES1_TST_CFG)

The SERDES 1 test configuration register (AIF_SERDES1_TST_CFG) is shown in Figure 89 and described in Table 75.

Figure 89. SERDES 1 Test Configuration Register (AIF_SERDES1_TST_CFG)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8
Reserved	INVPATT	Rate		Reserved			
R-0	R/W-0	R/W-0		R-0			

7	6	5	4	3	2	1	0
Loopback		CLK_BYP		EN_RX_PATT	EN_TX_PATT	TEST_PATT	
R/W-0		R/W-0		R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 75. SERDES 1 Test Configuration Register (AIF_SERDES1_TST_CFG) Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	INVPATT		Inverts polarity of pattern generator
13-12	Rate	0-3h	Selects rate for TX pattern generator
11-8	Reserved	0	Reserved
7-6	Loopback		Selects internal or bump/pad loopback
5-4	CLK_BYP	0-3h	Facilitates bypassing of the PLL clock and bypassing of the recovered receive clock
3	EN_RX_PATT		Enables generation of test patterns in the receiver
2	EN_TX_PATT		Enables generation of test patterns in the transmitter
1-0	TEST_PATT	0-3h	Test pattern selects one of two PRBS or two clock test patterns

8.2.4 Link Configuration (LINK_CFG) Link 0-5 Registers

The link configuration registers (LINK_CFG) for links 0-5 are shown in Figure 90 and described in Table 76.

Figure 90. Link Configuration (LINK_CFG) Registers

31	30	29	28	27	26	24
Reserved		SD_TX_TEST_EN	SD_RX_TEST_EN	Reserved	SD_TX_DE	
R-0		R/W-0	R/W-0	R-0	R/W-0	

23	22	20	19	18	17	16
SD_TX_DE	SD_TX_SWING		SD_TX_INV_TX_PAIR	SD_TX_CM	SD_RX_EQ	
R/W-0	R/W-0		R/W-0	R/W-0	R/W-0	

15	14	13	11	10	8
SD_RX_EQ		SD_RX_CDR		SD_RX_TERM	
R/W-0		R/W-0		R/W-0	

7	6	5	4	3	2	1	0
Reserved	SD_RX_INV _RX_PAIR	SD_RX_LOS_EN	SD_RX _ALIGN_EN	LINK_RATE		TX_LINK_EN	RX_LINK_EN
R-0	R/W-0	R/W-0	R/W-0	R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 76. Link Configuration Registers (LINK_CFG) Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29	TX_TEST_EN		Enables test modes specified via TESTCFG for this transmitter (1)
		0	Disabled
		1	Enabled
28	RX_TEST_EN		Enables test modes specified via TESTCFG for this receiver (1)
		0	Disabled
		1	Enabled
27	Reserved	0	Reserved
26-23	TX_DE		Selects one of 15 output de-emphasis settings from 4.76 to 71.42% (1)
		0000	0
		0001	4.76
		0010	9.52
		0011	14.28
		0100	19.04
		0101	23.8
		0110	28.56
		0111	33.32
		1000	38.08
		1001	42.85
		1010	47.61
		1011	52.38
		1100	57.14
		1101	61.9
		1110	66.66
		1111	71.42

(1) See TMS320TCI6487/8 SERDES Hardware Design Guidelines

Table 76. Link Configuration Registers (LINK_CFG) Field Descriptions (continued)

Bit	Field	Value	Description
22-20	TX_SWING		Selects one of eight output amplitude settings between 125 and 1250mVdfpp (1)
		000	125
		001	250
		010	500
		011	625
		100	750
		101	1000
		110	1250
		111	1375
19	TX_INV_TX_PAIR		Inverts polarity of TXPi and TXNi (1)
		0	Normal polarity;TXPi considered to be positive data and TXNi negative
		1	Inverted polarity; TXPi considered to be negative data and TXNi positive
18	SD_TX_CM		Adjusts the common mode to suit the termination at the attached receiver (1)
		0	Normal common mode; common mode not adjusted
		1	Raised common mode; common mode is raised by 5% of output amplitude
17-14	SD_RX_EQ		Enables and configures the adaptive equalizer to compensate for loss in the transmission media
		Value	**Low Freq Gain** / **Zero Freq (Mhz)**
		0000	Maximum / -
		0001	Adaptive
		0010	Reserved
		0011	Reserved
		0100	Reserved
		0101	Reserved
		0110	Reserved
		0111	Reserved
		1000	Adaptive / 365
		1001	Adaptive / 275
		1010	Adaptive / 195
		1011	Adaptive / 140
		1100	Adaptive / 105
		1001	Adaptive / 75
		1010	Adaptive / 55
		1111	Adaptive / 50
13-11	SD_RX_CDR		Configures the clock/data recovery algorithm (1)
		000	First order, threshold of 1
		001	First order, threshold of 16
		010	Second order, high precision, threshold of 1
		011	Second order, high precision, threshold of 16
		100	Second order, low precision, threshold of 1
		101	Second order, low precision, threshold of 16
		110	First order, threshold of 1 with fast lock
		111	Second order, low precision with fast lock

Table 76. Link Configuration Registers (LINK_CFG) Field Descriptions (continued)

Bit	Field	Value	Description
10-8	SD_RX_TERM		Selects input termination options suitable for variety of AC or DC coupled scenarios (1)
		000	DC coupled systems using CML transmitters
		001	AC coupled systems
		010	Reserved
		011	DC coupled systems that require the common mode voltage to be determined by the transmitter only
		1xx	Reserved
7	Reserved	0	Reserved
6	SD_RX_INV_RX_PAIR		Inverts polarity of RXPi and RXNi (1)
		0	Normal Polarity; RXPi considered to be positive data and RXNi negative
		1	Inverted Polarity; RXPi considered to be negative data and RXNi positive
5	SD_RX_LOS_EN		Enable SERDES loss of signal
		0	Disabled
		1	Enabled
4	SD_RX_ALIGN_EN		Enables internal or external symbol alignment.
		0	Alignment disabled
		1	Comma alignment enabled
3-2	LINK_RATE		Select both RX and TX line rate per link
		00	4x
		01	2x
		10	1x
		11	Reserved
1	TX_LINK_EN		TX link enable. Affects SERDES, Tx MAC, AG, CD, CO and PE
		0	Disabled
		1	Enabled
0	RX_LINK_EN		RX link enable. Affects SERDES, Rx MAC, CD, CI, and PD
		0	Disabled
		1	Enabled

There are six of these registers, one for each link:

LINK0_CFG	LINK2_CFG	LINK4_CFG
LINK1_CFG	LINK3_CFG	LINK5_CFG

8.2.5 Rx MAC Configuration Registers

There are six sets of the Rx MAC link configuration, pi offset, and loss of signal threshold registers, one set for each link.

8.2.5.1 Rx MAC Link Configuration (RM_LINK_CFG) Registers

The Rx MAC link configuration registers for links 0-5 are shown in Figure 91 and described in Table 77.

Figure 91. Rx MAC Link Configuration (RM_LINK_CFG) Registers

31 28	27 16
Reserved	RM_RCVD_MSTR_FRAME_WIND
R-0	R/W-0

15 8	7 6	5	4	3 1	0
RM_VALID_MSTR_FRAME_WIND	Reserved	RM_EXTRA_K28P7_ ERR_SUPPRESS	RM_ERROR_ SUPPRESS	RM_FORCE_ RX_STATE	RM_RECEIVER _EN
R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 77. Rx MAC Link Configuration (RM_LINK_CFG) Registers Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	RM_RCVD_MSTR_FRAME_WIND	0-0FFFh	Defines the measurement window for received master frame offset, given in the number of VBUS_CLK ticks. A value of 0x00 disables frame alignment checking. Range: 1 to 4095 VBUS_CLK tics.
15-8	RM_VALID_MSTR_FRAME_WIND	0-07Fh	Defines the measurement window for valid master frame offset, given in the number of VBUS_CLK tics. Related to the error: MSTR_FRAME_BNDRY_OUT_OF_RANGE. A value of 0x00 disables frame alignment checking. Range: 1 to 255 VBUS_CLK tics.
7-6	Reserved	0	Reserved
5	RM_EXTRA_K28P7_ERR_SUPPRESS		Suppress error reporting of extra K28.7 characters detected in the data stream. RP3 requires that messages be rejected when master frame boundary is detected outside the allowed window.
		0	Allow error reporting of extra K28.7 characters
		1	Suppress error reporting of extra K28.7 characters
4	RM_ERROR_SUPPRESS		Suppress error reporting when the receiver state machine is not in state ST3.
		0	Allow all Rx MAC error reporting when ST3
		1	Suppress all Rx MAC error reporting when ST3
3-1	RM_FORCE_RX_STATE		Force receiver state machine state
		000	Not active
		100	Force ST0 state
		101	Force ST1 state
		110	Force ST2 state
		111	Force ST3 state
0	RM_RECEIVER_EN		Rx MAC
		0	Disable
		1	Enable

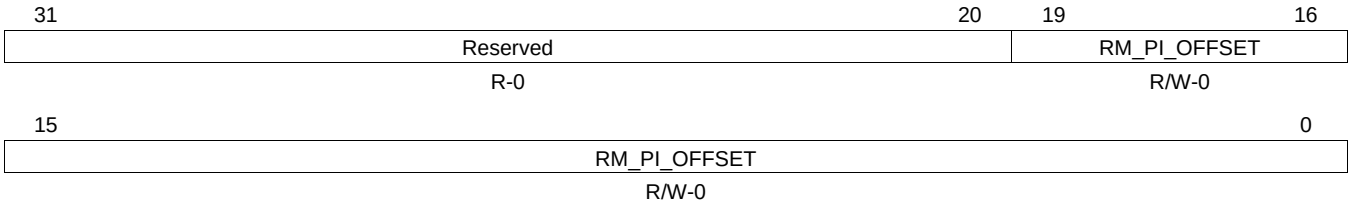
There are six of these registers, one for each link:

RM_LINK0_CFG	RM_LINK2_CFG	RM_LINK4_CFG
RM_LINK1_CFG	RM_LINK3_CFG	RM_LINK5_CFG

8.2.5.2 Rx MAC Links 0-5 Pi Offset Registers (RM_LINK_PI_OFFSET_CFG)

The Rx MAC links 0-5 Pi offset registers (RM_LINK_PI_OFFSET_CFG) is shown in Figure 92 and described in Table 78.

Figure 92. Rx MAC Links 0-5 Pi Offset Registers (RM_LINK_PI_OFFSET_CFG)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 78. Rx MAC Links 0-5 Pi Offset Registers (RM_LINK_PI_OFFSET_CFG) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	RM_PI_OFFSET	0-0x7FFFF	A 2's complement number specifying the delay from frame sync to the expected master frame / hyper-frame boundary (OBSAI RP3: K28.7 or CPRI: K28.5 AND HFN = 149) within ~65 ns (about 5, 1x link byte clocks) and given in the number of VBUS_CLK tics. Range: OBSAI: +/- 2^{19} - 1, CPRI: +2^{19} - 1 VBUS_CLK tics

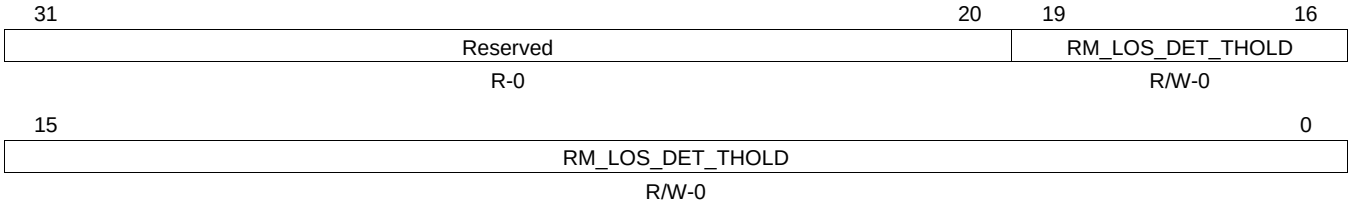
There are six of these registers, one for each link:

RM_LINK0_PI_OFFSET_CFG	RM_LINK2_PI_OFFSET_CFG	RM_LINK4_PI_OFFSET_CFG
RM_LINK1_PI_OFFSET_CFG	RM_LINK3_PI_OFFSET_CFG	RM_LINK5_PI_OFFSET_CFG

8.2.5.3 Rx MAC Links 0-5 LOS Threshold Registers (RM_LINK_LOS_THOLD_CFG)

The Rx MAC links 0-5 LOS threshold registers (RM_LINK_LOS_THOLD_CFG) are shown in Figure 93 and described in Table 79.

Figure 93. Rx MAC Links 0-5 LOS Threshold Registers (RM_LINK_LOS_THOLD_CFG)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 79. Rx MAC Links 0-5 LOS Threshold Registers (RM_LINK_LOS_THOLD_CFG) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	RM_LOS_DET_THOLD	0-FFFFFh	Sets 8b10b LOS detect threshold values in the number of line code violations received. Writing to this location automatically clears the RM_NUM_LOS counter and the RM_LOS_DET status bit.

There are six of these registers, one for each link:

RM_LINK0_LOS_THOLD_CFG	RM_LINK2_LOS_THOLD_CFG	RM_LINK4_LOS_THOLD_CFG
RM_LINK1_LOS_THOLD_CFG	RM_LINK3_LOS_THOLD_CFG	RM_LINK5_LOS_THOLD_CFG

8.2.5.4 Rx MAC Common Sync Counter (RM_SYNC_CNT_CFG) Register

The Rx MAC common sync counter (RM_SYNC_CNT_CFG) is shown in Figure 94 and described in Table 80.

Figure 94. Rx MAC Common Sync Counter (RM_SYNC_CNT_CFG) Register

31	16
RM_FRAME_SYNC_T	
R/W-0	

15	0
RM_SYNC_T	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 80. Rx MAC Common Sync Counter (RM_SYNC_CNT_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-16	RM_FRAME_SYNC_T	0-FFFFh	Threshold value for consecutive valid message groups which result in state ST3 (range: 0 to 65,535).
15-0	RM_SYNC_T	0-FFFFh	Threshold value for consecutive valid blocks of bytes which result in state ST1 (range: 0 to 65,535).

8.2.5.5 Rx MAC Common UnSync Counter (RM_UNSYNC_CNT_CFG) Register

The Rx MAC common unsync counter (RM_UNSYNC_CNT_CFG) is shown in register Figure 95 and described in Table 81.

Figure 95. Rx MAC Common UnSync Counter (RM_UNSYNC_CNT_CFG)Register

31	16
RM_FRAME_UNSYNC_T	
R/W-0	

15	0
RM_UNSYNC_T	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 81. Rx MAC Common UnSync Counter (RM_UNSYNC_CNT_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-16	RM_FRAME_UNSYNC_T	0-FFFFh	Threshold value for consecutive invalid message groups that result in the ST1 state (range: 0 to 65,535).
15-0	RM_UNSYNC_T	0-FFFFh	Threshold value for consecutive invalid blocks of bytes that result in ST0 state (range: 0 to 65,535).

8.2.6 Tx MAC Configuration Registers

There are six sets of the Tx MAC registers, one for each link.

Multiple back-to-back writes to the same Tx MAC MMR configuration register is discouraged, since it may cause the Tx MAC to operate erratically due to the use of multiple clock domains within the Tx MAC. Always allow at least eight VBUS_CLK cycles between writes to the same TM register.

8.2.6.1 Tx MAC Links 0-5 Configuration Registers 0 (TM_LINK_CFGA)

The Tx MAC links 0-5 configuration registers 0 for CPRI only (TM_LINK_CFGA) are shown in Figure 96 and described in Table 82.

Figure 96. Tx MAC Links 0-5 Configuration Registers 0 for CPRI Only (TM_LINK_CFGA)

31	16
TM_DELTA_OFFSET	
R/W-0	

15 13	12 8	7 6	5	4	3 1	0
Reserved	TM_FIFO_FULL_THOLD	Reserved	TM_LOS_EN	TM_FRAME_XMIT_EN	TM_FORCE_TX_STATE	TM_TRANSMITTER_EN
R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 82. Tx MAC Links 0-5 Configuration Registers 0 for CPRI only (TM_LINK_CFGA) Field Descriptions

Bit	Field	Value	Description
31-16	TM_DELTA_OFFSET	0-7FFFh	A 2's complement number specifying the delay from frame sync to start of master frame in BYTE_CLK tick increments (range: OBSAI: +/- 2^{15} - 1, CPRI: +2^{15} - 1 BYTE_CLK ticks).
15-13	Reserved	0	Reserved
12-8	TM_FIFO_FULL_THOLD	10-20	Sets the threshold for the full indication flag for the Tx MAC FIFO (range: 1 to 31 bytes). Program TM_FIFO_FULL_THOLD on the low side of thE 10-20 range for redirection, on the high side for decombining and somewhere in the low of the middle for combining.
7-6	Reserved	0	Reserved
5	TM_LOS_EN		Enable the impact of the RM_LOSS_OF_SIGNAL to the transmitter state machine
		0	Disable
		1	Enable
4	TM_FRAME_XMIT_EN		Allowing the TX FSM to enter state ST2. This bit has no effect if the state machine is already in state ST2.
		0	Disable
		1	Enable
3-1	TM_FORCE_TX_STATE		Force transmitter state machine state
		000	Not active
		100	Force ST0 state
		101	Force ST1 state
		110	Force ST2 state
		111	Not defined
0	TM_TRANSMITTER_EN		Tx MAC enable
		0	Disable
		1	Enable

There are six of these registers, one for each link:

TM_LINK0_CFGA	TM_LINK2_CFGA	TM_LINK4_CFGA
TM_LINK1_CFGA	TM_LINK3_CFGA	TM_LINK5_CFGA

8.2.6.2 Tx MAC Links 0-5 Configuration Registers 1 - CPRI only (TM_LINK_CFGB)

The Tx MAC links 0-5 configuration registers 1 - CPRI only (TM_LINK_CFGB) is shown in Figure 97 and described in Table 83.

Figure 97. Tx MAC Links 0-5 Configuration Registers 1 - CPRI only (TM_LINK_CFGB)

31	24	23	16
TM_PNTR_P		TM_LINK_CF	
R/W-0		R/W-0	

15	8	7	5	4	3	2	1	0
TM_START_UP		Reserved		TM_RCVD_LOF	TM_RCVD_LOS	TM_RCVD_SDI	TM_RCVD_RAI	TM_RCVD_RST
R/W-0		R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 83. Tx MAC Link 0-5 Configuration Registers 1 - CPRI only (TM_LINK_CFGB) Field Descriptions

Bit	Field	Value	Description
31-24	TM_PNTR_P	0-FFh	Transmit Pointer P
23-16	TM_LINK_CF	0-FFh	Transmit start-up information (Z.66.0)
15-8	TM_START_UP	0-FFh	Transmit protocol version (Z.2.0)
7-5	Reserved	0	Reserved
4	TM_RCVD_LOF		Transmit LOF (Z.130.0, b4)
		0	Do not transmit
		1	Transmit
3	TM_RCVD_LOS		Transmit LOS (Z.130.0, b3)
		0	Do not transmit
		1	Transmit
2	TM_RCVD_SDI		Transmit SDI (Z.130.0, b2)
		0	Do not transmit
		1	Transmit
1	TM_RCVD_RAI		Transmit RAI (Z.130.0, b1)
		0	Do not transmit
		1	Transmit
0	TM_RCVD_RST		Transmit reset (Z.130.0, b0)
		0	Do not transmit
		1	Transmit

There are six of these registers, one for each link:

TM_LINK0_CFGB	TM_LINK2_CFGB	TM_LINK4_CFGB
TM_LINK1_CFGB	TM_LINK3_CFGB	TM_LINK5_CFGB

8.2.6.3 Tx MAC Links 0-5 Configuration (TM_LINK_CFGC) Registers 2

The Tx MAC links 0-5 configuration (TM_LINK_CFGC) registers 2 are shown in Figure 98 and described in Table 84.

Figure 98. Tx MAC Links 0-5 Configuration (TM_LINK_CFGC) Registers 2

31	16
Reserved	
R-0	

15	8	7	0
TM_BFN_HIGH		TM_BFN_LOW	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 84. Tx MAC Links 0-5 Configuration (TM_LINK_CFGC) Registers 2 Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-8	TM_BFN_HIGH	0-FFh	Transmit node B frame number high byte (Z.130.0) - loads U_CNTR high byte
7-0	TM_BFN_LOW	0-FFh	Transmit node B frame number low byte (Z.128.0) - loads U_CNTR low byte

There are six of these registers, one for each link:

TM_LINK0_CFGC	TM_LINK2_CFGC	TM_LINK4_CFGC
TM_LINK1_CFGC	TM_LINK3_CFGC	TM_LINK5_CFGC

Note: Programming CPRI-only configuration bits has no effect in OBSAI RP3 mode.

8.2.7 Combiner / Decombiner Configuration Registers

Use the configuration register (CD_0CFG) to select which input link to route to the respective output link and to enable that input link. If you are redirecting one link to another, the values 0 through 5 represent the input link number to be re-routed. If combining, the value six selects the output of combiner 0 and the value 7 selects the output of combiner 1. If you are decombining, each output link mux source select that you use for decombining should select the same input link as the decombiner source select (DC0_SRC_SEL or DC1_SRC_SEL). The input link must be enabled for any of the CD operations (redirection, combining, or decombining).

8.2.7.1 Combiner / Decombiner Output Mux Select Configuration (CD_OUT_MUX_SEL_CFG) Register

The combiner / decombiner output mux select configuration (CD_OUT_MUX_SEL_CFG) register is shown in Figure 99 and described in Table 85.

Figure 99. Combiner / Decombiner Output Mux Select Configuration (CD_OUT_MUX_SEL_CFG) Register

31 28	27	26	25	24	23	22 20	19	18 16
Reserved	DC1_EN	DC0_EN	CB1_EN	CB0_EN	CD_OUT5_EN	CD_OUT_MUX5_SRC_SEL	CD_OUT4_EN	CD_OUT_MUX4_SRC_SEL
R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14 12	11	10 8	7	6 4	3	2 0
CD_OUT3_EN	CD_OUT_MUX3_SRC_SEL	CD_OUT2_EN	CD_OUT_MUX2_SRC_SEL	CD_OUT1_EN	CD_OUT_MUX1_SRC_SEL	CD_OUT0_EN	CD_OUT_MUX0_SRC_SEL
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 85. Combiner / Decombiner Output Mux Select Configuration (CD_OUT_MUX_SEL_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27	DC1_EN		Decombiner 1 enable
		0	Disable
		1	Enable
26	DC0_EN		Decombiner 1 enable
		0	Disable
		1	Enable
25	CB1_EN		Combiner 1 enable
		0	Disable
		1	Enable
24	CB0_EN		Combiner 1 enable
		0	Disable
		1	Enable
23	CD_OUT5_EN		CD input 5 enable
		0	Disable
		1	Enable input link 5 of the combiner / decombiner.
22-20	CD_OUT_MUX5_SRC_SEL	0-7h	Select source of link 5 output
19	CD_OUT4_EN		CD input 4 enable
		0	Disable
		1	Enable input link 4 of the combiner / decombiner.
18-16	CD_OUT_MUX4_SRC_SEL	0-7h	Select source of link 4 output

Table 85. Combiner / Decombiner Output Mux Select Configuration (CD_OUT_MUX_SEL_CFG) Register Field Descriptions (continued)

Bit	Field	Value	Description
15	CD_OUT3_EN		CD input 3 enable
		0	Disable
		1	Enable input link 3 of the combiner / decombiner.
14-12	CD_OUT_MUX3_SRC_SEL	0-7h	Select source of link 3 output
11	CD_OUT2_EN		CD input 2 enable
		0	Disable
		1	Enable input link 2 of the combiner / decombiner.
10-8	CD_OUT_MUX2_SRC_SEL	0-7h	Select source of link 2 output
7	CD_OUT1_EN		CD input 1 enable
		0	Disable
		1	Enable input link 1 of the combiner / decombiner.
6-4	CD_OUT_MUX1_SRC_SEL	0-7h	Select source of link 1 output
3	CD_OUT0_EN		CD input 0 enable
		0	Disable
		1	Enable input link 0 of the combiner / decombiner.
2-0	CD_OUT_MUX0_SRC_SEL		Select source of link 0 output:
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	CB0
		111	CB1

8.2.7.2 Combiner Source Select Configuration (CD_CB_SRC_SEL_CFG) Register

The combiner source select register provides a four location table for each of the two combiners. Each entry in the table describes the source link to be combined. Entry A is the first link to be combined, entry B is the second link to be combined, etc. The list repeats in round-robin fashion. The table must always be complete since the combiner cycles through the entire table. If there are fewer links to combine than can fill the output link, use empty links as table entries.

The combiner source select configuration (CD_CB_SRC_SEL_CFG) register is shown in Figure 100 and described in Table 86.

Figure 100. Combiner Source Select Configuration (CD_CB_SRC_SEL_CFG) Register

31	30 28	27	26 24	23	22 20	19	18 16
Reserved	CB1_SRC_D_SEL	Reserved	CB1_SRC_C_SEL	Reserved	CB1_SRC_B_SEL	Reserved	CB1_SRC_A_SEL
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14 12	11	10 8	7	6 4	3	2 0
Reserved	CB0_SRC_D_SEL	Reserved	CB0_SRC_C_SEL	Reserved	CB0_SRC_B_SEL	Reserved	CB0_SRC_A_SEL
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

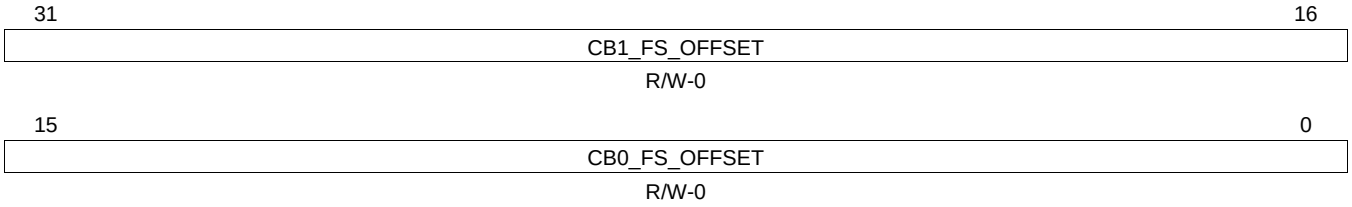
Table 86. Combiner Source Select Configuration (CD_CB_SRC_SEL_CFG) Register Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	CB1_SRC_D_SEL	0-7h	Combiner 1 input source link select table entry D.
27	Reserved	0	Reserved
26-24	CB1_SRC_C_SEL	0-7h	Combiner 1 input source link select table entry C.
23	Reserved	0	Reserved
22-20	CB1_SRC_B_SEL	0-7h	Combiner 1 input source link select table entry B.
19	Reserved	0	Reserved
18-16	CB1_SRC_A_SEL	0-7h	Combiner 1 input source link select table entry A.
15	Reserved	0	Reserved
14-12	CB0_SRC_D_SEL	0-7h	Combiner 1 input source link select table entry D.
11	Reserved	0	Reserved
10-8	CB0_SRC_C_SEL	0-7h	Combiner 0 input source link select table entry C.
7	Reserved	0	Reserved
6-4	CB0_SRC_B_SEL	0-7h	Combiner 0 input source link select table entry B.
3	Reserved	0	Reserved
2-0	CB0_SRC_A_SEL		Combiner 0 input source link select table entry A.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty

8.2.7.3 Combiner Alignment Offset Configuration (CD_CB_OFFSET_CFG) Register

The combiner alignment offset Configuration (CD_CB_OFFSET_CFG) register is shown in Figure 101 and described in Table 87.

Figure 101. Combiner Alignment Offset Configuration (CD_CB_OFFSET_CFG) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 87. Combiner Alignment Offset Configuration (CD_CB_OFFSET_CFG) Register Field Descriptions

Bit	Field	Value	Description [(1)]
31-16	CB1_FS_OFFSET	0-FFFFh	Combiner 1 offset from frame sync. Similar to Rx MAC PI_OFFSET. This value is used to check input link frame alignment. The value is given in VBUS_CLK tics. Range = 0 to 65,535 (i.e., positive values only).
15-0	CB0_FS_OFFSET	0-FFFFh	Combiner 0 offset from frame sync. Similar to Rx MAC pi_offset. This value is used to check input link frame alignment. The value is given in VBUS_CLK tics. Range = 0 to 65,535 (i.e., positive values only).

(1) For combiner offset calculations, see Section A.1.

8.2.7.4 Combiner Valid Window Configuration (CD_CB_VALID_WIND_CFG) Register

The combiner valid window configuration (CD_CB_VALID_WIND_CFG) register is shown in Figure 102 and described in Table 88.

Figure 102. Combiner Valid Window Configuration (CD_CB_VALID_WIND_CFG) Register

31	16
Reserved	
R-0	

15	13	12	8	7	5	4	0
Reserved		CB1_VALID_DATA_WIND		Reserved		CB0_VALID_DATA_WIND	
R-0		R/W-0		R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 88. Combiner Valid Window Configuration (CD_CB_VALID_WIND_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12-8	CB1_VALID_DATA_WIND	0-1Fh	Combiner 1 valid data window for master frame offset, in VBUS_CLK ticks (27 × 409.6 MHz VBUS_CLK = 65 ns). Range values 0 to 31 create a window 1 to 32 VBUS_CLK cycles long. Values typically range from 12 to 26.
7-5	Reserved	0	Reserved
4-0	CB0_VALID_DATA_WIND	0-1Fh	Combiner 0 valid data window for master frame offset, in VBUS_CLK ticks (27 × 409.6 MHz VBUS_CLK = 65 ns). Range: values 0 to 31 create a window 1 to 32 VBUS_CLK cycles long. Values typically range from 12 to 26.

8.2.7.5 Decombiner Source Select Configuration (CD_DC_SRC_SEL_CFG) Register

The link source selected in the Decombiner source select register must match what was selected in the output multiplexer register (CD_OUT_MUX_SRC_SEL). The decombiner source select configuration (CD_DC_SRC_SEL_CFG) register is shown in Figure 103 and described in Table 89.

Figure 103. Decombiner Source Select Configuration (CD_DC_SRC_SEL_CFG) Register

31	16
Reserved	
R-0	

15	7	6	4	3	2	0
Reserved		DC1_SRC_SEL		Reserved	DC0_SRC_SEL	
R-0		R/W-0		R-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 89. Decombiner Source Select Configuration (CD_DC_SRC_SEL_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	DC1_SRC_SEL		Decombiner 1 source (i.e., select 1 of 6 links to be decombined as follows:
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Undefined
		111	Undefined
3	Reserved	0	Reserved
2-0	DC0_SRC_SEL		Decombiner 0 source (i.e., select 1 of 6 links to decombine)
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Undefined
		111	Undefined

8.2.7.6 Decombiner Destination Select Configuration (CD_DC_DST_SEL_CFG) Register

The decombiner destination select configuration register provides a 4-location table for each of the two decombiners. Each entry in the table describes a destination link. Entry 'A' is the first destination link for data from the source link, entry 'B' the second, etc. The list repeats in round-robin fashion. The table must always be complete since the decombiner cycles through the entire table. If there are fewer destination links for decombining than originated from the source link, use 'empty' links as table entries.

The decombiner destination select configuration (CD_DC_DST_SEL_CFG) register is shown in Figure 104 and described in Table 90.

Figure 104. Decombiner Destination Select Configuration (CD_DC_DST_SEL_CFG) Register

31	30 28	27	26 24	23	22 20	19	18 16
Reserved	DC1_DEST_D_SEL	Reserved	DC1_DEST_C_SEL	Reserved	DC1_DEST_B_SEL	Reserved	DC1_DEST_A_SEL
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14 12	11	10 8	7	6 4	3	2 0
Reserved	DC0_DEST_D_SEL	Reserved	DC0_DEST_C_SEL	Reserved	DC0_DEST_B_SEL	Reserved	DC0_DEST_A_SEL
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 90. Decombiner Destination Select Configuration (CD_DC_DST_SEL_CFG) Register Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	DC1_DEST_D_SEL	0-7h	Decombiner 1 output destination link select table entry 'D'.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
27	Reserved	0	Reserved
26-24	DC1_DEST_C_SEL	0-7h	Decombiner 1 output destination link select table entry 'C'.
		000	Link 0
		001	Link 1
		010	LInk 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
23	Reserved	0	Reserved

Table 90. Decombiner Destination Select Configuration (CD_DC_DST_SEL_CFG) Register Field Descriptions (continued)

Bit	Field	Value	Description
22-20	DC1_DEST_B_SEL	0-7h	Decombiner 1 output destination link select table entry ‘B’.
		000	Link 0
		001	Link 1
		010	Link 2
		011	LInk 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
19	Reserved	0	Reserved
18-16	DC1_DEST_A_SEL	0-7h	Decombiner 1 output destination link select table entry ‘A’.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
15	Reserved	0	Reserved
14-12	DC0_DEST_D_SEL	0-7h	Decombiner 0 output destination link select table entry ‘D’.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
11	Reserved	0	Reserved
10-8	DC0_DEST_C_SEL	0-7h	Decombiner 0 output destination link select table entry ‘C’.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
7	Reserved	0	Reserved

Table 90. Decombiner Destination Select Configuration (CD_DC_DST_SEL_CFG) Register Field Descriptions (continued)

Bit	Field	Value	Description
6-4	DC0_DEST_B_SEL	0-7h	Decombiner 0 output destination link select table entry 'B'.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty
3	Reserved	0	Reserved
2-0	DC0_DEST_A_SEL	0-7h	Decombiner 0 output destination link select table entry 'A'.
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Empty
		111	Empty

8.2.8 Aggregator Configuration Registers Links 0-5 (AG_LINK_CFG)

The aggregator configuration registers links 0-5 (AG_LINK_CFG) are shown in Figure 105 and described in Table 91.

Figure 105. Aggregator Configuration Registers Links 0-5 (AG_LINK_CFG)

31	16
Reserved	
R-0	

15	4	3	2	1	0
Reserved		HDR_ERR_SEL	ERR_EN	AG_MODE	
R-0		R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 91. Aggregator Configuration Registers Links 0-5 (AG_LINK_CFG) Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	HDR_ERR_SEL		This bit controls the data source selected by the aggregator when there is an OBSAI header mismatch and the PE is requesting aggregation. This bit has no effect for CPRI.
		0	CD
		1	PE
2	ERR_EN		Aggregator error-checking enable
		0	Aggregator error-checking disabled
		1	Aggregator error-checking enabled

Table 91. Aggregator Configuration Registers Links 0-5 (AG_LINK_CFG) Field Descriptions (continued)

Bit	Field	Value	Description
1-0	AG_MODE	0-3h	Select aggregator mode(1)
		00	Not enabled (no data sent to the Tx MAC)
		01	Pass through from combiner only
		10	Insertion from the protocol encoder only
		11	All aggregation functions are enabled.

(1) The PE signaling is ignored in mode 1, but a PE ADD command changes the operation of the AG in mode 2. Once a link is enabled, changes to AG_MODE require the link be disabled and then re-enabled for the AG to recognize the changes.

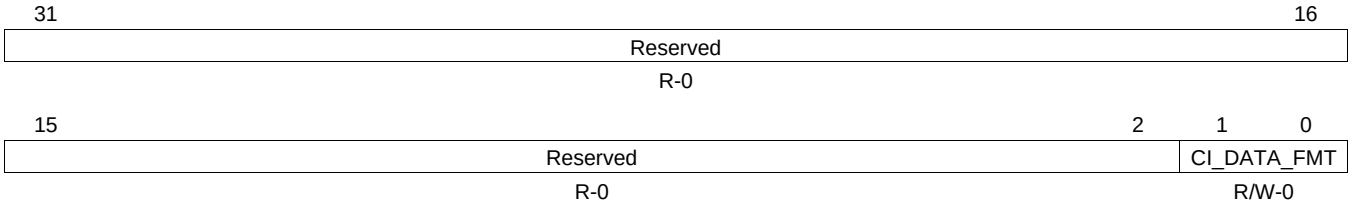
There are six of these registers, one for each link:

AG_LINK0_CFG	AG_LINK2_CFG	AG_LINK4_CFG
AG_LINK1_CFG	AG_LINK3_CFG	AG_LINK5_CFG

8.2.9 CPRI Input Data Format Converter Configuration Registers Links 0-5 (CI_LINK_CFG)

The CPRI input data format converter configuration registers links 0-5 (CI_LINK_CFG) are shown in Figure 106 and described in Table 92.

Figure 106. CPRI Input Data Format Converter Configuration Registers Links 0-5 (CI_LINK_CFG)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 92. CPRI Input Data Format Converter Configuration Registers Links 0-5 (CI_LINK_CFG) Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1-0	CI_DATA_FMT		CPRI data format
		00	UL/PIC 7
		01	UL/PIC 8-bit
		10	DL/Generic 15-bit
		11	DL/Generic 16-bit

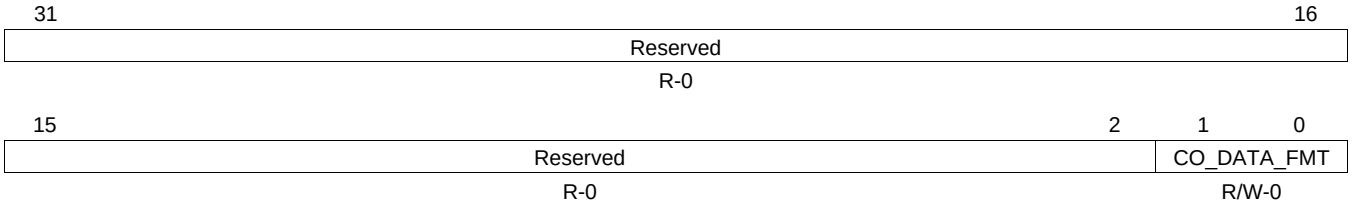
There are six of these registers, one for each link:

CI_LINK0_CFG	CI_LINK2_CFG	CI_LINK4_CFG
CI_LINK1_CFG	CI_LINK3_CFG	CI_LINK5_CFG

8.2.10 CPRI Output Data Format Converter Configuration (CO_LINK_CFG) Registers Links 0-5

The CPRI output data format converter configuration (CO_LINK_CFG) registers links 0-5 is shown in Figure 107 and described in Table 93.

Figure 107. CPRI Output Data Format Converter Configuration (CO_LINK_CFG) Registers Links 0-5



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 93. CPRI Output Data Format Converter Configuration (CO_LINK_CFG) Registers Links 0-5 Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1-0	CO_DATA_FMT		CPRI data format:
		00	UL/PIC 7-bit
		01	UL/PIC 8-bit
		10	DL/generic 15-bit
		11	DL/generic 16-bit

There are six of these registers, one for each link:

CO_LINK0_CFG	CO_LINK2_CFG	CO_LINK4_CFG
CO_LINK1_CFG	CO_LINK3_CFG	CO_LINK5_CFG

8.2.11 Protocol Decoder Configuration Registers

8.2.11.1 Protocol Decoder Configuration (PD_0_CFG) Register

The protocol decoder configuration (PD_0_CFG) register is shown in Figure 108 and described in Table 94.

Figure 108. Protocol Decoder Configuration (PD_0_CFG) Register

31	28	27	16
Reserved		CPRI_DATA_FORMAT	
R-0		R/W-0	

15	14	13	8	7	6	5	0
Reserved		PD_LINK_BURST_8VS4		Reserved		PD_LINK_EN	
R-0		R/W-0		R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 94. Protocol Decoder Configuration (PD_0_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-28	Reserved	0	Reserved
27-16	CPRI_DATA_FORMAT		Bit-mapped 2 bits per link, CPRI internal data format: 2'b. In CPRI mode, there are not sufficient cycles to arbitrate the address LUT for this purpose.
		00	DL format (only legal with burst of 4) 2'b
		01	UL RSA format 2'b
		10	PIC MAI 2'b
		11	Generic format
15-14	Reserved	0	Reserved
13-8	PD_LINK_BURST_8VS4		Bit-mapped per link 1'b
		0	Burst of four links (such as in DL format: this only applies to circuit-switched messages).
		1	Burst of eight links (such as in UL RSA format) 1'b
7-6	Reserved	0	Reserved
5-0	PD_LINK_EN		Bit-mapped per link
		0	Link disabled
		1	Link enabled

8.2.11.2 Protocol Decoder Configuration (PD_1_CFG) Register

The protocol decoder configuration (PD_1_CFG) register is shown in Figure 109 and described in Table 95.

Figure 109. Protocol Decoder Configuration (PD_1_CFG) Register

31	30	29	24	23	18	17	16
Reserved		CPRI_PKTSW_EN		Reserved		PD_OFFSET_ADR	
R-0		R/W-0		R-0		R/W-0	

15	0
PD_OFFSET_ADR	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 95. Protocol Decoder Configuration (PD_1_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-24	CPRI_PKTSW_EN	3Fh	Bit-mapped per link. Enables CPRI control word capture (using PKTSW RAM)
23-18	Reserved	0	Reserved
17-0	PD_OFFSET_ADR	0-3FFFFh	Bit-mapped three bits per link. Circular offsets write to circuit-switched RAM by n×4 chips. Eight-chip-orientated UL data disregards the LSB, assuming it is zero. This offset is used to correct RAM alignment between streams that different MOD32 offsets have with one another. Allows chained DMAs to be more regular.

8.2.11.3 Protocol Decoder OBSAI Adr Mux Select (PD_ADR_MUX_SEL_CFG) Register

The protocol decoder OBSAI ADR MUX select (PD_ADR_MUX_SEL_CFG) register is shown in Figure 110 and described in Table 96.

Figure 110. Protocol Decoder OBSAI Adr Mux Select (PD_ADR_MUX_SEL_CFG) Register

31	20	19	18	17	16
Reserved		PD_ADR_MUX9_CTRL		PD_ADR_MUX8_CTRL	
R-0		R/W-0		R/W-0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PD_ADR_MUX7_CTRL		PD_ADR_MUX6_CTRL		PD_ADR_MUX5_CTRL		PD_ADR_MUX4_CTRL		PD_ADR_MUX3_CTRL		PD_ADR_MUX2_CTRL		PD_ADR_MUX1_CTRL		PD_ADR_MUX0_CTRL	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 96. Protocol Decoder OBSAI Adr Mux Select (PD_ADR_MUX_SEL_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-18	PD_ADR_MUX9_CTRL		Select address 9 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
17-16	PD_ADR_MUX8_CTRL		Select address 8 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
15-14	PD_ADR_MUX7_CTRL		Select address 7 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
13-12	PD_ADR_MUX6_CTRL		Select address 6 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
11-10	PD_ADR_MUX5_CTRL		Select address 5 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
9-8	PD_ADR_MUX4_CTRL		Select address 4 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]

(1) Bit-mapped two bits per supported address bit.

Table 96. Protocol Decoder OBSAI Adr Mux Select (PD_ADR_MUX_SEL_CFG) Register Field Descriptions (continued)

Bit	Field	Value	Description
7-6	PD_ADR_MUX3_CTRL		Select address 3 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
5-4	PD_ADR_MUX2_CTRL		Select address 2 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
3-2	PD_ADR_MUX1_CTRL		Select address 1 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]
1-0	PD_ADR_MUX0_CTRL		Select address 0 mux as address LUT index.(1)
		00	Use ADR[n+0]
		01	Use ADR[n+1]
		10	Use ADR[n+2]
		11	Use ADR[n+3]

8.2.11.4 Protocol Decoder Type CirSw Capture Enable LUT (PD_TYPE_CIR_LUT_CFG) Register

The protocol decoder type CirSW capture enable LUT (PD_TYPE_CIR_LUT_CFG) register is shown in Figure 111 and described in Table 97.

Figure 111. Protocol Decoder Type CirSw Capture Enable LUT (PD_TYPE_CIR_LUT_CFG) Register

31	16
CIR_LUT_CFG_PD_TYPE_LUT_CIR_SW	
R/W-0	

15	0
CIR_LUT_CFG_PD_TYPE_LUT_CIR_SW	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 97. Protocol Decoder Type CirSw Capture Enable LUT (PD_TYPE_CIR_LUT_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-0	CIR_LUT_CFG_PD_TYPE_LUT_CIR_SW	0-FFFF FFFFh	Bit mapped per 0-to-31 OBASI type: enable circuit-switched, capture to circuit-switched RAM 1'b
		0	Discard
		1	Capture 1'b

8.2.11.5 Protocol Decoder Type PktSw Capture Enable LUT (PD_TYPE_PKT_LUT_CFG) Register

The protocol decoder type PktSw capture enable LUT (PD_TYPE_PKT_LUT_CFG) register is shown in Figure 112 and described in Table 98.

Figure 112. Protocol Decoder Type PktSw Capture Enable LUT (PD_TYPE_PKT_LUT_CFG) Register

31	16
PD_TYPE_LUT_PKT_SW	
R/W-0	

15	0
PD_TYPE_LUT_PKT_SW	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

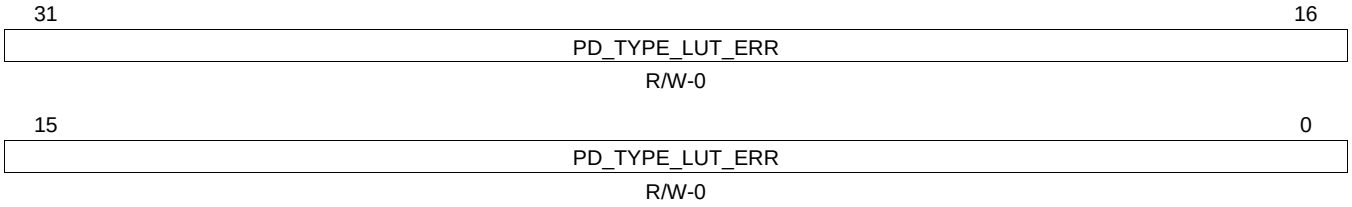
Table 98. Protocol Decoder Type PktSw Capture Enable LUT (PD_TYPE_PKT_LUT_CFG) Register

Bit	Field	Value	Description
31-0	PD_TYPE_LUT_PKT_SW	0-FFFF FFFFh	Bit-mapped per 0-to-31 OBASI type: enable circuit-switched capture to circuit-switched RAM
		0	Discard
		1	Capture

8.2.11.6 Protocol Decoder Type Error (PD_TYPE_ERR_LUT_CFG) Register

The protocol decoder type error (PD_TYPE_ERR_LUT_CFG) register is shown in Figure 113 and described in Table 99.

Figure 113. Protocol Decoder Type Error (PD_TYPE_ERR_LUT_CFG) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 99. Protocol Decoder Type (PD_TYPE_ERR_LUT_CFG) Error Register

Bit	Field	Value	Description
31-0	PD_TYPE_LUT_ERR	0-FFFF FFFFh	Bit mapped per 0-to-31 OBASI type:
		0	No Error
		1	Illegal OBSAI type, any inbound messages with this type cause an error/alarm condition and trigger error FIFO capture of message.

8.2.11.7 Protocol Decoder Address Look-Up Table (PD_ADR_LUT) Register

The protocol decoder address look-up table (PD_ADR_LUT) register is shown in Figure 114 and described in Table 100. This address LUT is 1K deep and shared between all six PD links.

Figure 114. Protocol Decoder Address Look-Up Table (PD_ADR_LUT) Register

31	20	19	18	17	16
Reserved		DATA_FORMAT1_LUT		AXC_INDEX1_LUT	
R-0		R/W-0		R/W-0	

15	14	13	12	11	10	9	8	7	4	3	2	1	0
AXC_INDEX1_LUT		PKT_FIFO_INDEX1_LUT		AXC_CAPT_EN1_LUT	BAD_ADR1_LUT	DATA_FORMAT0_LUT		AXC_INDEX0_LUT		PKT_FIFO_INDEX0_LUT		AXC_CAPT_EN0_LUT	BAD_ADR0_LUT
R/W-0		R/W-0		R/W-0	R/W-0	R/W-0		R/W-0		R/W-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 100. Protocol Decoder Address Look-Up Table (PD_ADR_LUT) Register Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-18	DATA_FORMAT1_LUT		Odd addresses: bits [1:0]: internal data format: 2'b
		00	DL format (only legal with burst of 4) 2'b
		01	UL RSA format (only legal with burst of 8) 2'b
		10	PIC MAI 2'b
		11	Generic format (1)
17-14	AXC_INDEX1_LUT	0-Fh	Odd addresses: bits [3:0]: antenna carrier index, 0-15 used for address generation for circuit-switched types (unused for packet-switched types)
13-12	PKT_FIFO_INDEX1_LUT		Odd addresses: bits [1:0]: FIFO index (used for packet-switched type), 0-3 used to direct message to one of four FIFOs.
		00	0x0-to-0x2: packet-switched FIFOs
		01	0x0-to-0x2: packet-switched FIFOs
		10	0x0-to-0x2: packet-switched FIFOs
		11	0x3: Error FIFO
11	AXC_CAPT_EN1_LUT		Odd addresses: 1'b1 causes circuit-switched type to also capture error FIFO (FIFO index 0x3)
		0	No action.
		1	Capture error FIFO.
10	BAD_ADR1_LUT		Odd addresses: Indicates illegal address: 1'b.
		0	No action
		1	Error condition is triggered marking address as illegal.
9-8	DATA_FORMAT0_LUT		Even addresses: bits [1:0]: internal data format: 2'b
		00	DL format (only legal with burst of 4) 2'b
		01	UL RSA format (only legal with burst of 8) 2'b
		10	PIC MAI
		11	Generic format (1)
7-4	AXC_INDEX0_LUT	0-Fh	Even addresses: bits [3:0]: antenna carrier index, 0-15 used for address generation for circuit-switched types (unused for packet-switched types)

(1) OBSAI Pkt_switched format is assumed for PKT_switched FIFO.

Table 100. Protocol Decoder Address Look-Up Table (PD_ADR_LUT) Register Field Descriptions (continued)

Bit	Field	Value	Description
3-2	PKT_FIFO_INDEX0_LUT		Even addresses: bits [1:0]: FIFO index (used for packet-switched type), 0-3 used to direct message to one of four FIFOs.
		00	0x0-to-0x2: packet-switched FIFOs
		01	0x0-to-0x2: packet-switched FIFOs
		10	0x0-to-0x2: packet-switched FIFOs
		11	0x3: Error FIFO
1	AXC_CAPT_EN0_LUT		Even addresses: 1'b1 causes circuit-switched type to also capture error FIFO (FIFO index 0x3)
		0	No action.
		1	Capture error FIFO.
0	BAD_ADR0_LUT		Even addresses: Indicates illegal address: 1'b.
		0	No action.
		1	Error condition is triggered marking address as illegal.

8.2.11.8 Protocol Decoder 84 Count Look-Up Table Bits [31:0] (PD_LINK_84CNT_LUT0_CFG) Registers Links 0-5

The protocol decoder 84 count look-up table bits [31:0] (PD_LINK_84CNT_LUT0_CFG) registers links 0-5 are shown in Figure 115 and described in Table 101.

Figure 115. Protocol Decoder 84 Count Look-Up Table Bits [31:0] (PD_LINK_84CNT_LUT0_CFG) Registers Links 0-5

31		16
TIME_STAMP_INCR_NXT_LUT0		
R/W-0		

15		0
TIME_STAMP_INCR_NXT_LUT0		
R/W-0		

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 101. Protocol Decoder 84 Count Look-Up Table Bits [31:0] (PD_LINK_84CNT_LUT0_CFG) Registers Links 0-5 Field Descriptions

Bit	Field	Value	Description
31-0	TIME_STAMP_INCR_NXT_LUT0	0-FFFF FFFFh	Bit-mapped by 84 count. Controls time stamp to increment, incremented value is used for next message. (Only used for OBSAI, CPRI TS increments every basic frame)
		0	Hold current count
		1	Increment

There are six of these registers, one for each link:

PD_LINK0_84CNT_LUT0_CFG	PD_LINK2_84CNT_LUT0_CFG	PD_LINK4_84CNT_LUT0_CFG
PD_LINK1_84CNT_LUT0_CFG	PD_LINK3_84CNT_LUT0_CFG	PD_LINK5_84CNT_LUT0_CFG

8.2.11.9 Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT1_CFG) Register Bits [63:32]

The protocol decoder 84 count look-up table (PD_LINK_84CNT_LUT1_CFG) register bits [63:32] is shown in Figure 116 and described in Table 102.

Figure 116. Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT0_CFG) Register bits [63:32]

31	16
TIME_STAMP_INCR_NXT_LUT1	
R/W-0	

15	0
TIME_STAMP_INCR_NXT_LUT1	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 102. Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT1_CFG) Register bits [63:32]

Bit	Field	Value	Description
31-0	TIME_STAMP_INCR_NXT_LUT1	0-FFFF FFFFh	Bits [31:0], bit-mapped by 84 count. Controls time stamp to increment, incremented value is used for next message. (Only used for OBSAI, CPRI TS increments every basic frame) 1'b
		0	Hold current count
		1	Increment 1'b

There are six of these registers, one for each link:

PD_LINK0_84CNT_LUT1_CFG	PD_LINK2_84CNT_LUT1_CFG	PD_LINK4_84CNT_LUT1_CFG
PD_LINK1_84CNT_LUT1_CFG	PD_LINK3_84CNT_LUT1_CFG	PD_LINK5_84CNT_LUT1_CFG

8.2.11.10 Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT2_CFG) Register bits [83:64]

The protocol decoder 84 count look-up table (PD_LINK_84CNT_LUT0_CFG) register bits [83:64] is shown in Figure 117 and described in Table 103.

TIME_STAMP_INCR_NXT_LUT2 should be set to 1 for OBSAI. In CPRI, this register in not used and may be set to any value. The hardware has special handling of frame boundaries; expected time stamps are zeroed and the burst_start indicator to the trackers are forced. When programming this particular bit to 1 the hardware-forced frame boundary condition is more consistent with the programmed operation.

Figure 117. Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT2_CFG) Register bits [83:64]

31	20	19	16
Reserved		TIME_STAMP_INCR_NXT_LUT2	
R-0		R/W-0	

15	0
TIME_STAMP_INCR_NXT_LUT2	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 103. Protocol Decoder 84 Count Look-Up Table (PD_LINK_84CNT_LUT2_CFG) Register bits [83:64]

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	TIME_STAMP_INCR_NXT_LUT2	0-FFFFFh	Bits [83:64], bit-mapped by 84 count. Controls time stamp to increment, incremented value is used for current operation 1'b
		0	Hold current count
		1	Increment 1'b

There are six of these registers, one for each link:

PD_LINK0_84CNT_LUT2_CFG	PD_LINK2_84CNT_LUT2_CFG	PD_LINK4_84CNT_LUT2_CFG
PD_LINK1_84CNT_LUT2_CFG	PD_LINK3_84CNT_LUT2_CFG	PD_LINK5_84CNT_LUT2_CFG

8.2.11.11 Protocol Decoder CPRI Stream Index LUT0 (PD_LINK_CPRI_SI_LUT0_CFG) Register

The protocol decoder CPRI stream index (PD_LINK_CPRI_SI_LUT0_CFG) register is shown in Figure 118 and described in Table 104.

Figure 118. Protocol Decoder CPRI Stream Index LUT0 (PD_LINK_CPRI_SI_LUT0_CFG) Register

31 28	27 24	23 20	19 16
CPRI_SI_AXC_7_LUT	CPRI_SI_AXC_6_LUT	CPRI_SI_AXC_5_LUT	CPRI_SI_AXC_4_LUT
R/W-0	R/W-0	R/W-0	R/W-0

15 12	11 8	7 4	3 0
CPRI_SI_AXC_3_LUT	CPRI_SI_AXC_2_LUT	CPRI_SI_AXC_1_LUT	CPRI_SI_AXC_0_LUT
R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 104. Protocol Decoder CPRI Stream Index LUT0 Register (PD_LINK_CPRI_SI_LUT0_CFG) Field Descriptions

Bit	Field	Value	Description [(1)]
31-28	CPRI_SI_AXC_7_LUT	0-Fh	AxC 7, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
27-24	CPRI_SI_AXC_6_LUT	0-Fh	AxC 6, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
23-20	CPRI_SI_AXC_5_LUT	0-Fh	AxC 5, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
19-16	CPRI_SI_AXC_4_LUT	0-Fh	AxC 4, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
15-12	CPRI_SI_AXC_3_LUT	0-Fh	AxC 3, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
11-8	CPRI_SI_AXC_2_LUT	0-Fh	AxC 2, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
7-4	CPRI_SI_AXC_1_LUT	0-Fh	AxC 1, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
3-0	CPRI_SI_AXC_0_LUT	0-Fh	AxC 0, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.

(1) In OBSAI mode, address_LUT serves the purpose of the AxC value.

There are six of these registers, one for each link:

PD_LINK0_CPRI_SI_LUT0_CFG PD_LINK2_CPRI_SI_LUT0_CFG PD_LINK4_CPRI_SI_LUT0_CFG
PD_LINK1_CPRI_SI_LUT0_CFG PD_LINK3_CPRI_SI_LUT0_CFG PD_LINK5_CPRI_SI_LUT0_CFG

8.2.11.12 Protocol Decoder CPRI Stream Index LUT1 (PD_LINK_CPRI_SI_LUT1_CFG) Register

The protocol decoder CPRI stream index (PD_LINK_CPRI_SI_LUT1_CFG) Register is shown in Figure 119 and described in Table 105.

Figure 119. Protocol Decoder CPRI Stream Index LUT 1 (LUT1 PD_LINK_CPRI_SI_LUT1_CFG) Register

31 28	27 24	23 20	19 16
CPRI_SI_AXC_7_LUT1	CPRI_SI_AXC_6_LUT1	CPRI_SI_AXC_5_LUT1	CPRI_SI_AXC_4_LUT1
R/W-0	R/W-0	R/W-0	R/W-0
15 12	**11 8**	**7 4**	**3 0**
CPRI_SI_AXC_3_LUT1	CPRI_SI_AXC_2_LUT1	CPRI_SI_AXC_1_LUT1	CPRI_SI_AXC_0_LUT1
R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 105. Protocol Decoder CPRI Stream Index LUT 1 (PD_LINK_CPRI_SI_LUT1_CFG) Register

Bit	Field	Value	Description [(1)]
31-28	CPRI_SI_AXC_7_LUT1	0-Fh	AxC 15, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
27-24	CPRI_SI_AXC_6_LUT1	0-Fh	AxC 14, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
23-20	CPRI_SI_AXC_5_LUT1	0-Fh	AxC 13, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
19-16	CPRI_SI_AXC_4_LUT1	0-Fh	AxC 12, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
15-12	CPRI_SI_AXC_3_LUT1	0-Fh	AxC 11, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
11-8	CPRI_SI_AXC_2_LUT1	0-Fh	AxC 10, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
7-4	CPRI_SI_AXC_1_LUT1	0-Fh	AxC 9, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.
3-0	CPRI_SI_AXC_0_LUT1	0-Fh	AxC 8, CPRI stream index (SI). 0-to-15 value corresponding to streams organized in circuit-switched RAM. Used for address generation. AxC value used to index LUT.

(1) In OBSAI mode, address_LUT serves the purpose of the AxC value.

There are six of these registers, one for each link:

PD_LINK0_CPRI_SI_LUT1_CFG PD_LINK2_CPRI_SI_LUT1_CFG PD_LINK4_CPRI_SI_LUT1_CFG
PD_LINK1_CPRI_SI_LUT1_CFG PD_LINK3_CPRI_SI_LUT1_CFG PD_LINK5_CPRI_SI_LUT1_CFG

8.2.12 Protocol Encoder Configuration Registers

8.2.12.1 Protocol Encoder Configuration Register (PE_CFG)

The protocol encoder configuration register (PE_CFG) is shown in Figure 120 and described in Table 106.

Several configuration parameters for the PE are set globally. OBSAI versus CPRI mode is set in the AIF global register. Link rate is set in the transmit link configuration register. The endianess is set through VBUS signals.

Figure 120. Protocol Encoder Configuration (PE_CFG) Register

31	22	21	16
Reserved		LINK_BURST_8VS4	
R-0		R/W-0	

15	14	13	8	7	6	5	0
Reserved		LINK_ENABLE		Reserved		PE_CPRI_CNTL_EN	
R-0		R/W-0		R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

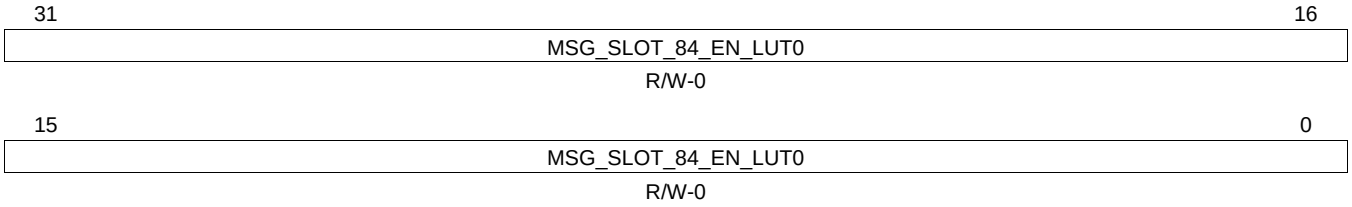
Table 106. Protocol Encoder Configuration (PE_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-16	LINK_BURST_8VS4		Bit-mapped per link 1'b
		0	Burst of four link (such as in DL format) (only applies to circuit-switched messages)
		1	Burst of eight link (such as in UL RSA format) 1'b
15-14	Reserved	0	Reserved
13-8	LINK_ENABLE		Bit-mapped per link
		0	Link disabled
		1	Link enabled
7-6	Reserved	0	Reserved
5-0	PE_CPRI_CNTL_EN		Bit-mapped per link 1'b
		0	CPRI control words are assumed to be zero (expect for K character)
		1	CPRI words are read from RAM and constructed 1'b

8.2.12.2 Protocol Encoder 84 Count Message Enable Bits [31-0] (PE_LINK_84_EN_LUT0_CFG) Register

The protocol encoder 84 count message enable bits [31-0] (PE_LINK_84_EN_LUT0_CFG) register is shown in Figure 121 and described in Table 107.

Figure 121. Protocol Encoder 84 Count Message Enable Bits [31-0] (PE_LINK_84_EN_LUT0_CFG)



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 107. Protocol Encoder 84 Count Message Enable Bits [31-0] (PE_LINK_84CNT_LUT0_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-0	MSG_SLOT_84_EN_LUT0	0-FFFF FFFFh	Bits [31:0], bit-mapped by 84 count. Enable of a particular message slot.
		0	Disable message slot, insert empty message (double-buffered to take effect on frame boundary)
		1	Enable message slot for inserted message 1'b

There are six of these registers, one for each link:

PE_LINK0_84_EN_LUT0_CFG	PE_LINK2_84_EN_LUT0_CFG	PE_LINK4_84_EN_LUT0_CFG
PE_LINK1_84_EN_LUT0_CFG	PE_LINK3_84_EN_LUT0_CFG	PE_LINK5_84_EN_LUT0_CFG

8.2.12.3 Protocol Encoder 84 Count Message Enable Bits [63-32] (PE_LINK_84_EN_LUT1_CFG) Register

The protocol encoder 84 count message enable bits [63-32] (PE_LINK_84_EN_LUT1_CFG) register is shown in Figure 122 and described in Table 108.

Figure 122. Protocol Encoder 84 Count Message Enable Bits [63-32] (PE_LINK_84_EN_LUT1_CFG) register

31	16
MSG_SLOT_84_EN_LUT1	
R/W-0	

15	0
MSG_SLOT_84_EN_LUT1	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 108. Protocol Encoder 84 Count Message Enable Bits [63-32] (PE_LINK_84_EN_LUT1_CFG) register Field Descriptions

Bit	Field	Value	Description
31-0	MSG_SLOT_84_EN_LUT1	0-FFFF FFFFh	Bits [63:32]: Bit mapped by 84 count. Enable of a particular message slot. 1'b
		0	Disable message slot, insert empty msg (Double buffered to take effect on frame boundary)
		1	Enable message slot for inserted msg

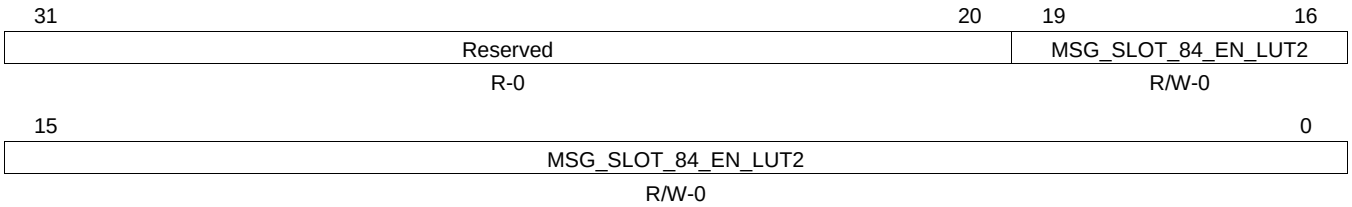
There are six of these registers, one for each link:

PE_LINK0_84_EN_LUT1_CFG	PE_LINK2_84_EN_LUT1_CFG	PE_LINK4_84_EN_LUT1_CFG
PE_LINK1_84_EN_LUT1_CFG	PE_LINK3_84_EN_LUT1_CFG	PE_LINK5_84_EN_LUT1_CFG

8.2.12.4 Protocol Encoder 84 Count message Enable bits [83:64] (PE_LINK_84_EN_LUT2_CFG) Register

The protocol encoder 84 count message enable bits [83:64] (PE_LINK_84_EN_LUT2_CFG) register is shown in Figure 123 and described in Table 109.

Figure 123. Protocol Encoder 84 Count message Enable bits [83:64] (PE_LINK_84_EN_LUT2_CFG) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 109. Protocol Encoder 84 Count message Enable bits [83:64] (PE_LINK_84_EN_LUT2_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved
19-0	MSG_SLOT_84_EN_LUT2	0-0000 FFFFh	Bits [83:64]: Bit-mapped by 84 count. Enable of a particular message slot. Dou 1'b
		0	Disable message slot, insert empty message (double-buffered to take effect on frame boundary)
		1	Enable message slot for inserted msg 1'b

There are six of these registers, one for each link:

PE_LINK0_84_EN_LUT2_CFG	PE_LINK2_84_EN_LUT2_CFG	PE_LINK4_84_EN_LUT2_CFG
PE_LINK1_84_EN_LUT2_CFG	PE_LINK3_84_EN_LUT2_CFG	PE_LINK5_84_EN_LUT2_CFG

8.2.12.5 Protocol Encoder Transmission Rule Terminal Count 0 & 1 (PE_LINK_TERM_CNT01_CFG) Register

The protocol encoder transmission rule terminal count 0 & 1 (PE_TERM_CNT01_CFG) register is shown in Figure 124 and described in Table 110.

Figure 124. Protocol Encoder Transmission Rule Terminal Count 0 & 1 (PE_LINK_TERM_CNT01_CFG) Register

31	27	26	16
Reserved		CTRL_MSG_TERM_CNT1	
R-0		R/W-0	

15	11	10	0
Reserved		CTRL_MSG_TERM_CNT0	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 110. Protocol Encoder Transmission Rule Terminal Count 0 & 1 (PE_LINK_TERM_CNT01_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26-16	CTRL_MSG_TERM_CNT1	0-7FFh	Sets the terminal count for the Control Message, Transmission Rule counter, Counter #1
15-11	Reserved	0	Reserved
10-0	CTRL_MSG_TERM_CNT0	0-7FFh	Sets the terminal count for the Control Message, Transmission Rule counter, Counter #0

There are six of these registers, one for each link:

PE_LINK0_TERM_CNT01_CFG	PE_LINK2_TERM_CNT01_CFG	PE_LINK4_TERM_CNT01_CFG
PE_LINK1_TERM_CNT01_CFG	PE_LINK3_TERM_CNT01_CFG	PE_LINK5_TERM_CNT01_CFG

8.2.12.6 Protocol Encoder Transmission Rule Terminal Count 2 & 3 (PE_LINK_TERM_CNT23_CFG) Register

The protocol encoder transmission rule terminal count 2 & 3 (PE_TERM_CNT23_CFG) register is shown in Figure 125 and described in Table 111.

Figure 125. Protocol Encoder Transmission Rule Terminal Count 2 & 3 (PE_LINK_TERM_CNT23_CFG) Register

31	27	26	16
Reserved		CTRL_MSG_TERM_CNT3	
R-0		R/W-0	

15	11	10	0
Reserved		CTRL_MSG_TERM_CNT2	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 111. Protocol Encoder Transmission Rule Terminal Count 2 & 3 (PE_LINK_TERM_CNT23_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26-16	CTRL_MSG_TERM_CNT3	0-7FFh	Sets the terminal count for the Control Message, Transmission Rule counter, Counter #3
15-11	Reserved	0	Reserved
10-0	CTRL_MSG_TERM_CNT2	0-7FFh	Sets the terminal count for the Control Message, Transmission Rule counter, Counter #2

There are six of these registers, one for each link:

PE_LINK0_TERM_CNT23_CFG	PE_LINK2_TERM_CNT23_CFG	PE_LINK4_TERM_CNT23_CFG
PE_LINK1_TERM_CNT23_CFG	PE_LINK3_TERM_CNT23_CFG	PE_LINK5_TERM_CNT23_CFG

8.2.12.7 Protocol Encoder 84 Count LUT (PE_LINK_84CNT_LUT) Register

The protocol encoder 84 count LUT (PE_LINK_84CNT_LUT) register provides an 84-element look up table. You must access the full width of the table. You cannot use partial-byte lane writes. This register is not used for CPRI.

The PE_LINK_84CNT_LUT register is shown in Figure 126 and described in Table 112.

Figure 126. Protocol Encoder 84 Count LUT (PE_LINK_84CNT_LUT) Register

31	16
Reserved	
R-0	

15	11	10	9	7	6	5	4	0
Reserved		INCR_COUNT_NOW_LUT	SELECT_COUNT_LUT		TIME_STAMP_INCR_NXT_LUT	CKT_VS_PKT_SWTCH_MSG_LUT	AXC_OR_CTRL_INDEX_LUT	
R-0		R/W-0	R/W-0		R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 112. Protocol Encoder 84 Count LUT (PE_LINK_84CNT_LUT) Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	Reserved
10	INCR_COUNT_NOW_LUT		Controls counter to increment, incremented value is used for next operation (current operation uses current counter value).
		0	Does not increment the counter
		1	Increments the counter
9-7	SELECT_COUNT_LUT		Selects counter LUT
		0-3h	4x control slot counters
		4h	1x data slot counter for increment and for use with transmission rule
6	TIME_STAMP_INCR_NXT_LUT		Controls timestamp to increment, incremented value is used for next message.
		0	Does not increment timestamp
		1	Increments timestamp.
5	CKT_VS_PKT_SWTCH_MSG_LUT		Selects data type
		0	Packet-switched data
		1	Circuit-switched data
4-0	AXC_OR_CTRL_INDEX_LUT (1)	0-1Fh	Index for accessing the identity LUT
		0-15h	16 possible antenna carriers
		16-19h	Four possible streams of control slots
		20h	Packet-switched data in message slot

(1) Stream Index is equal to bits [3:0] of this field. SI is a four bit number which is used for circuit switched address generation. (SI is only valid for cirsw type)

There are six of these registers, one for each link:

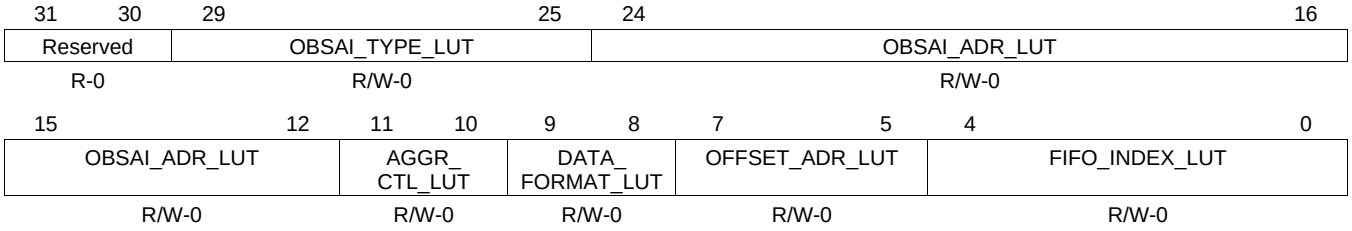
PE_LINK0_84CNT_LUT	PE_LINK2_84CNT_LUT	PE_LINK4_84CNT_LUT
PE_LINK1_84CNT_LUT	PE_LINK3_84CNT_LUT	PE_LINK5_84CNT_LUT

8.2.12.8 Protocol Encoder Identity LUT Part 0 (PE_LINK_ID_LUT0) Register

The protocol encoder identity LUT part 0 (PE_LINK_ID_LUT0) register provides a 21-element look up table. You must access the full width of the table. You cannot use partial-byte lane writes.

The PE_LINK_ID_LUT0 register is shown in Figure 127 and described in Table 113.

Figure 127. Protocol Encoder Identity LUT Part 0 (PE_LINK_ID_LUT0) Register



LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 113. Protocol Encoder Identity LUT Part 0 (PE_LINK_ID_LUT0) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-25	OBSAI_TYPE_LUT	0-Fh	OBSAI type, fixed per antenna carrier (circuit switched only)
24-12	OBSAI_ADR_LUT	0-FFFh	OBSAI address, fixed per antenna carrier (circuit switched only)
11-10	AGGR_CTL_LUT		Aggregation control bits (assuming Tx_Rule passes) 2'b:
		00	NOP (AG still pulls from FIFO, but does not use read PE data bytes)
		01	Insert (1)
		10	Add 15/16 bit
		11	Add 7/8 bit
9-8	DATA_FORMAT_LUT		Internal Data format: 2'b:
		00	DL Format(only legal with burst of 4)
		01	UL RSA Format(only legal for burst of 8)
		10	PIC MAI
		11	Generic Format (OBSAI Pkt_Switched format is assumed for PKT_SWITCHED FIFO)
7-5	OFFSET_ADR_LUT	0-7h	Circularly offsets reads from circuit switched RAM by n*4 chips. Eight-chip orientated UL data disregards the LSB, assuming it is zero. This offset is used to correct RAM alignment between streams different MOD32 offsets with one another. Allows chained DMAs to be more regular.
4-0	FIFO_INDEX_LUT	0-1Fh	PKTSW FIFO Index. Used to address FIFO corresponding to this link.

(1) Use insert for header information only. For payload, always use add. For the first TMS320TCI6487/8 in the Daisy Chain, program the AG for PE-only mode.

There are six of these registers, one for each link:

PE_LINK0_ID_LUT0	PE_LINK2_ID_LUT0	PE_LINK4_ID_LUT0
PE_LINK1_ID_LUT0	PE_LINK3_ID_LUT0	PE_LINK5_ID_LUT0

8.2.12.9 Protocol Encoder Identity LUT part 1 (PE_LINK_ID_LUT1) Register

The protocol encoder identity LUT part 1 (PE_LINK_ID_LUT1) register provides the second portion of the 21-element identity LUT. The identity LUT is split into the two parts because it is wider then the 32-bit VBUS configuration interface. You must access the full width of the table. You cannot use partial-byte lane write.

The PE_LINK_ID_LUT1 register is shown in Figure 128 and described in Table 114.

Figure 128. Protocol Encoder Identity LUT part 1(PE_LINK_ID_LUT1) register

31	22	21	16
Reserved		TX_RULE_MCOMPARE_LUT	
R-0		R/W-0	

15	11	10	0
TX_RULE_MCOMPARE_LUT		TX_RULE_MASK_LUT	
R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 114. Protocol Encoder Identity LUT part 1 (PE_LINK_ID_LUT1) Register Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-11	TX_RULE_MCOMPARE_LUT	0-7FFh	Transmission rule compare bits. Field is bitwise XNOR with current counter value. Enabled results (see TX_RULE_MASK) are reduction and giving a pass/fail of the transmission rule.
10-0	TX_RULE_MASK_LUT		Transmission rule mask. Indicates which counter bits should be compared.
		0	Indicates that the bit automatically passes compare.
		1	Indicates that the counter bit should be compared with TX_RULE_COMPARE field.

There are six of these registers, one for each link:

PE_LINK0_ID_LUT1	PE_LINK2_ID_LUT1	PE_LINK4_ID_LUT1
PE_LINK1_ID_LUT1	PE_LINK3_ID_LUT1	PE_LINK5_ID_LUT1

8.2.13 Data Buffer Configuration Registers

8.2.13.1 Data Buffer Configuration (DB_GENERIC_CFG) Register

The data buffer configuration (DB_GENERIC_CFG) register is shown in Figure 129 and described in Table 115.

Figure 129. Data Buffer Configuration (DB_GENERIC_CFG) Register

31	15
Reserved	
R-0	

14	13	1	0
CAPTURE_EN	Reserved		MEM_LEAK_FLUSH
R/W-0	R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 115. Data Buffer Configuration (DB_GENERIC_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-15	Reserved	0	Reserved
14	CAPTURE_EN	1	Enables logic analyzer capture
13-1	Reserved	0	Reserved
0	MEM_LEAK_FLUSH (1)		Flushes outbound PKTSW FIFO's if a memory leak is detected. 1'b
		0	NOP
		1	Flushes all outbound PKTSW FIFO's.

(1) Corresponds to DB_PKTSW_OUT_MEM_LEAK EE event/status, see Section 8.3.7.4.

8.2.13.2 Data Buffer DMA Queue Clear (DB_DMA_QUE_CLR_CFG) Register

The data buffer DMA depth clear (DB_DMA_QUE_CLR_CFG) register provides manual means by which software can clear the 0-8 depth and burst queue-tracking counters. These counters all have hardware-automated means by which they are cleared (under certain error conditions).

The DB_DMA_QUE_CLR_CFG register is shown in Figure 130 and described in Table 116.

Figure 130. Data Buffer DMA Depth Clear (DB_DMA_QUE_CLR_CFG) Register

31	22	21	16
Reserved		OUT_DMA_PE_QUE_CLR	
R-0		R-0	

15	14	13	8	7	6	5	0
Reserved		OUT_DMA_VBUS_QUE_CLR		Reserved		IN_DMA_VBUS_QUE_CLR	
R-0		W-0		R-0		W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 116. Data Buffer DMA Depth Clear (DB_DMA_QUE_CLR_CFG) Register Field Descriptions

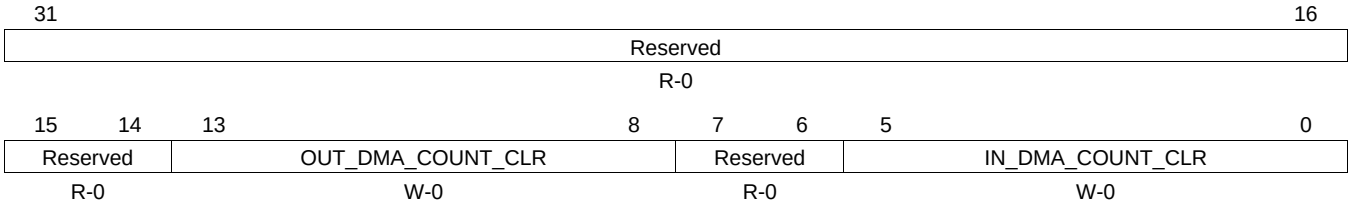
Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-16	OUT_DMA_PE_QUE_CLR		Bit-mapped per link, clears the outbound DMA PE Queue counters 1'b
		0	NOP
		1	Clears (acts as a strobe) 1'b
15-14	Reserved	0	Reserved
13-8	OUT_DMA_VBUS_QUE_CLR		Bit-mapped per link, clears the outbound DMA VBUS queue counters 1'b
		0	NOP
		1	Clears (acts as a strobe) 1'b
7-6	Reserved	0	Reserved
5-0	IN_DMA_VBUS_QUE_CLR (1)		Bit-mapped per link, clears the inbound DMA VBUS queue counters 1'b
		0	NOP
		1	Clears (acts as a strobe) 1'b

(1) The inbound DMA depth tracking counter has certain usage corner cases where the automated hardware-clearing mechanism cannot regain proper synchronization with the DMA engine. Under these corner cases, the software can manually force the alignment through these clear registers.

8.2.13.3 Data Buffer DMA Depth Clear (DB_DMA_CNT_CLR_CFG) Register

The data buffer DMA depth clear (DB_DMA_CNT_CLR_CFG) register is shown in Figure 131 and described in Table 117.

Figure 131. Data Buffer DMA Depth Clear (DB_DMA_CNT_CLR_CFG) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 117. Data Buffer DMA Depth Clear (DB_DMA_CNT_CLR_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved
13-8	OUT_DMA_COUNT_CLR		Bit-mapped per link, clears the outbound DMA depth counters 1'b
		0	NOP
		1	Clear (acts as a strobe) 1'b
7-6	Reserved	0	Reserved
5-0	IN_DMA_COUNT_CLR		Bit-mapped per link, clears the inbound DMA depth counters 1'b
		0	NOP
		1	Clear (acts as a strobe) 1'b

8.2.13.4 Data Buffer Outbound Packet-Switched FIFO Enable (DB_OUT_PKTSW_EN_CFG) Register

The data buffer outbound packet-switched FIFO enable (DB_OUT_PKTSW_EN_CFG) register is shown in Figure 132 and described in Table 118.

Figure 132. Data Buffer Outbound Packet-Switched FIFO Enable (DB_OUT_PKTSW_EN_CFG) Register

31	30	29	16
Reserved		PKTSW_FIFO_EN	
R-0		R/W-0	

15	0
PKTSW_FIFO_EN	
R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 118. Data Buffer Outbound Packet-Switched FIFO Enable (DB_OUT_PKTSW_EN_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-0	PKTSW_FIFO_EN	0-3FFFF FFFh	Bit-mapped per channel, enables packet-switched outbound FIFOs 1'b
		0	FIFO disabled. FIFO is flushed and VBUS writes are rejected (not ADDR_ER).
		1	FIFO enabled. VBUS writes are accepted.

8.2.13.5 Data Buffer Inbound Packet-Switched FIFO Flush (DB_OUT_PKTSW_FLUSH_CFG) Register

The data buffer inbound packet-switched FIFO flush (DB_OUT_PKTSW_FLUSH_CFG) register is shown in Figure 133 and described in Table 119.

Figure 133. Data Buffer Inbound Packet-Switched FIFO Flush (DB_OUT_PKTSW_FLUSH_CFG) Register

31	30	29	16
Reserved		PKTSW_FIFO_FLUSH	
R-0		W-0	

15	0
PKTSW_FIFO_FLUSH	
W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 119. Data Buffer Inbound Packet-Switched FIFO Flush (DB_OUT_PKTSW_FLUSH_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-0	PKTSW_FIFO_FLUSH	0-3FFFF FFFh	Bit-mapped per channel, enables packet-switched outbound FIFOs. 1'b
		0	NOP
		1	Flush FIFO. FIFO pointers clear. The FIFO is enabled (acts as a strobe)

8.2.13.6 Data Buffer Inbound Packet-Switched FIFO Empty_n Event Depth (DB_IN_FIFO_EVNT_CFG) Register

The data buffer inbound packet-switched FIFO empty_n event depth (DB_IN_FIFO_EVNT_CFG) register is shown in Figure 134 and described in Table 120.

Figure 134. Data Buffer Inbound Packet-Switched FIFO Empty_n Event Depth (DB_IN_FIFO_EVNT_CFG) Register

31	16
Reserved	
R-0	

15 12	11 8	7 4	3 0
Reserved	PKTSW_FIFO2_EVNT_DEPTH	PKTSW_FIFO1_EVNT_DEPTH	PKTSW_FIFO0_EVNT_DEPTH
R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 120. Data Buffer Inbound Packet-Switched FIFO Empty_n Event Depth (DB_IN_FIFO_EVNT_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-12	Reserved	0	Reserved
11-8	PKTSW_FIFO2_EVNT_DEPTH		Inbound packet-switched FIFO #2, programmable modulo count of FIFO writes for each _PNE system event strobe.
		0000	1
		0001	2
		0010	3
		0011	4
		0100	5
		0101	6
		0110	7
		0111	8
		1000	9
		1001	10
		1010	11
		1011	12
		1100	13
		1101	14
		1110	15
		1111	16

Table 120. Data Buffer Inbound Packet-Switched FIFO Empty_n Event Depth (DB_IN_FIFO_EVNT_CFG) Register Field Descriptions (continued)

Bit	Field	Value	Description
7-4	PKTSW_FIFO1_EVNT_DEPTH		Inbound packet-switched FIFO #1, programmable modulo count of FIFO writes for each _PNE system event strobe.
		0000	1
		0001	2
		0010	3
		0011	4
		0100	5
		0101	6
		0110	7
		0111	8
		1000	9
		1001	10
		1010	11
		1011	12
		1100	13
		1101	14
		1110	15
		1111	16
3-0	PKTSW_FIFO0_EVNT_DEPTH		Inbound packet-switched FIFO #0, programmable modulo count of FIFO writes for each _PNE system event strobe.
		0000	1
		0001	2
		0010	3
		0011	4
		0100	5
		0101	6
		0110	7
		0111	8
		1000	9
		1001	10
		1010	11
		1011	12
		1100	13
		1101	14
		1110	15
		1111	16

8.2.13.7 Data Buffer Inbound Packet-Switched FIFO Depth (DB_IN_FIFO_SIZE_CFG) Register

The data buffer inbound packet-switched FIFO depth (DB_IN_FIFO_SIZE_CFG) register is shown in Figure 135 and described in Table 121.

Any writes to the DB_IN_FIFO_SIZE_CFG register cause all inbound packet-switched FIFOs to be cleared and new size parameters to take effect.

Figure 135. Data Buffer Inbound Packet-Switched FIFO Depth (DB_IN_FIFO_SIZE_CFG) Register

31	21	20	16
Reserved		PKTSW_FIFO3_START	
R-0		R/W-0	

15	13	12	8	7	5	4	0
Reserved		PKTSW_FIFO2_START		Reserved		PKTSW_FIFO1_START	
R-0		R/W-0		R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 121. Data Buffer Inbound Packet-Switched FIFO Depth (DB_IN_FIFO_SIZE_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-16	PKTSW_FIFO3_START	0-1Fh	Inbound packet-switched FIFO, programmable depth of FIFO. Sets the starting address of Error_FIFO and ending address (minus 1) of FIFO #2. (Value x16 equals RAM quadword address)
15-13	Reserved	0	Reserved
12-8	PKTSW_FIFO2_START	0-1Fh	Inbound packet-switched FIFO, programmable depth of FIFO. Sets the starting address of FIFO #2 and ending address (minus 1) of FIFO #1. (Value x16 equals RAM quadword address)
7-5	Reserved	0	Reserved
4-0	PKTSW_FIFO1_START	0-1Fh	Inbound packet-switched FIFO, programmable depth of FIFO. Sets the starting address of FIFO #1 and ending address (minus 1) of FIFO #0. (Value x16 equals RAM quadword address).

8.2.13.8 Data Buffer Force System Events (DB_FORCE_SYSEVENT_CFG) Register

The data buffer force system events (DB_FORCE_SYSEVENT_CFG) register is shown in Figure 136 and described in Table 122.

Figure 136. Data Buffer Force System Events (DB_FORCE_SYSEVENT_CFG) Register

31	16
Reserved	
R-0	

15	8	7	6	0
Reserved		SYS_EVENT_ FORCE_CAPTBUF	SYS_EVENT_FORCE_INFIFO	
R-0		W-0	W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 122. Data Buffer Force System Events (DB_FORCE_SYSEVENT_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	SYS_EVENT_FORCE_CAPTBUF	7h	Writing 1'b1 to bit location forces AI_EVENT_CAPT_BUF_NE to strobe active for one VBUS_CLK cycle
6-0	SYS_EVENT_FORCE_INFIFO		Writing 1'b1 to bit location forces system events to strobe active for one VBUS_CLK cycle.
		0	AI_EVENT_INPKTSW_FIFO0_NE
		2h	AI_EVENT_INPKTSW_FIFO1_NE
		4h	AI_EVENT_INPKTSW_FIFO2_NE
		8h	AI_EVENT_INPKTSW_FIFO3_NE
		Fh	AI_EVENT_INPKTSW_FIFO0_PNE
		10h	AI_EVENT_INPKTSW_FIFO1_PNE
		20h	AI_EVENT_INPKTSW_FIFO2_PNE

8.2.13.9 Data Buffer PE Tracker Auto Sync Control (DB_OUTB_TRK_AUTOSYNC_CFG) Register

The data buffer PE tracker auto sync control (DB_OUTB_TRK_AUTOSYNC_CFG) register is shown in Figure 137 and described in Table 123.

Figure 137. Data Buffer PE Tracker Auto Sync Control (DB_OUTB_TRK_AUTOSYNC_CFG) Register

31	30 28	27	26 24	23	22 20	19	18 16
Reserved	TRK5_OUTB_ SYNC_VALUE	Reserved	TRK4_OUTB_ SYNC_VALUE	Reserved	TRK3_OUTB_ SYNC_VALUE	Reserved	TRK2_OUTB_ SYNC_VALUE
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14 12	11	10 8	7 6	5 0
Reserved	TRK1_OUTB_ SYNC_VALUE	Reserved	TRK0_OUTB_ SYNC_VALUE	Reserved	TRK_OUTB_AUTO_SYNC_EN
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 123. Data Buffer PE Tracker Auto Sync Control (DB_OUTB_TRK_AUTOSYNC_CFG) Register Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	TRK5_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 5.
27	Reserved	0	Reserved
26-24	TRK4_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 4.
23	Reserved	0	Reserved
22-20	TRK3_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 3.
19	Reserved	0	Reserved
18-16	TRK2_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 2.
15	Reserved	0	Reserved
14-12	TRK1_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 1.
11	Reserved	0	Reserved
10-8	TRK0_ OUTB_SYNC_VALUE	0-7h	Start value for PE-DMA & PE-FS Tracker at time of PE starting frame boundary for Link 0.
7-6	Reserved	0	Reserved
5-0	TRK_ OUTB_AUTO_SYNC_EN	0-3Fh	Bit-mapped per link; enables auto-resynchronization of PE-DMA & PE-FS trackers

8.2.13.10 Data Buffer PD Tracker Auto Sync Control (DB_INB_TRK_AUTOSYNC_CFG) Register

The data buffer PD tracker auto sync control (DB_INB_TRK_AUTOSYNC_CFG) register is shown in Figure 138 and described in Table 124.

Figure 138. Data Buffer PD Tracker Auto Sync Control (DB_INB_TRK_AUTOSYNC_CFG) Register

31	30	28	27	26	24
Reserved	TRK5_ INB_SYNC_VALUE		Reserved	TRK4_INB_SYNC_VALUE	
R-0	R/W-0		R-0	R/W-0	

23	22	20	19	18	16
Reserved	TRK3_INB_SYNC_VALUE		Reserved	TRK2_INB_SYNC_VALUE	
R-0	R/W-0		R-0	R/W-0	

15	14	12	11	10	8
Reserved	TRK1_INB_SYNC_VALUE		Reserved	TRK0_INB_SYNC_VALUE	
R-0	R/W-0		R-0	R/W-0	

7	6	5	0
Reserved		TRK_INB_AUTO_SYNC_EN	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 124. Data Buffer PD Tracker Auto Sync Control (DB_INB_TRK_AUTOSYNC_CFG) Register Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved
30-28	TRK5_ INB_SYNC_VALUE	0-7h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 5.
27	Reserved	0	Reserved
26-24	TRK4_ INB_SYNC_VALUE	0-7h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 4.
23	Reserved	0	Reserved
22-20	TRK3_ INB_SYNC_VALUE	0-7h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 3.
19	Reserved	0	Reserved
18-16	TRK2_ INB_SYNC_VALUE	0-3h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 2.
15	Reserved	0	Reserved
14-12	TRK1_ INB_SYNC_VALUE	0-3h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 1.
11	Reserved	0	Reserved
10-8	TRK0_ INB_SYNC_VALUE	0-3h	Start value for PD-DMA Tracker at time of PE starting frame boundary for Link 0.
7-6	Reserved	0	Reserved
5-0	TRK_INB_AUTO_SYNC_EN	0-3Fh	Bit-mapped per link; enables auto-resynchronization of PD-DMA trackers.

8.2.14 Exception Event Handler Configuration Registers

8.2.14.1 EE Configuration (EE_CFG) Register

The EE configuration (EE_CFG) register is shown in Figure 139 and described in Table 125.

Figure 139. EE Configuration (EE_CFG) Register

31	16
Reserved	
R-0	

15	4	3	2	1	0
Reserved		AI_EVENT3_ MODE	AI_EVENT2_ MODE	AI_EVENT1_ MODE	AI_EVENT0_ MODE
R-0		R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 125. EE Configuration (EE_CFG) Register Field Descriptions

Bit	Field	Value	Description
31-4	Reserved	0	Reserved
3	AI_EVENT3_MODE		AI_EVENT[3] mode
		0	Error/alarm mode: Generates an event and then waits for the EE End of Interrupt register to be written with the AI_EVENT[3] address to allow the subsequent event. Used when AI_EVENT[3] is generating an error/alarm.
		1	Trigger mode: Generates an event on AI_EVENT[3] each time an internal AIF event occurs that is mapped to AI_EVENT[3]. Used when AI_EVENT[3] is generating a trigger.
2	AI_EVENT2_MODE		AI_EVENT[2] mode
		0	Error/alarm mode: Generates an event and then waits for the EE End of Interrupt register to be written with the AI_EVENT[2] address to allow the subsequent event. Used when AI_EVENT[2] is generating an error/alarm.
		1	Trigger mode: Generates an event on AI_EVENT[2] each time an internal AIF event occurs that is mapped to AI_EVENT[2]. Used when AI_EVENT[2] is generating a trigger.
1	AI_EVENT1_MODE		AI_EVENT[1] mode
		0	Error/alarm mode: Generates an event and then waits for the EE End of Interrupt register to be written with the AI_EVENT[1] address to allow the subsequent event. Used when AI_EVENT[1] is generating an error/alarm.
		1	Trigger mode: Generates an event on AI_EVENT[1] each time an internal AIF event occurs that is mapped to AI_EVENT[1]. Used when AI_EVENT[1] is generating a trigger.
0	AI_EVENT0_MODE		AI_EVENT[0] mode
		0	Error/alarm mode: Generates an event and then waits for the EE End of Interrupt register to be written with the AI_EVENT[0] address to allow the subsequent event. Used when AI_EVENT[0] is generating an error/alarm.
		1	Trigger mode: Generates an event on AI_EVENT[0] each time an internal AIF event occurs that is mapped to AI_EVENT[0]. Used when AI_EVENT[0] is generating a trigger.

8.2.14.2 EE Link Select Register A for AI_EVENT[2] (EE_LINK_SEL_EV2A)

The EE link select register A for AI_EVENT[2] (EE_LINK_SEL_EV2A) is shown in Figure 140 and described in Table 126.

Figure 140. EE Link Select Register A for AI_EVENT[2] (EE_LINK_SEL_EV2A)

31 27	26 24	23	22 20	19	18 16
Reserved	DB_LINK_SELECT	Reserved	CD_LINK_SELECT	Reserved	AG_LINK_SELECT
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

15	14 12	11	10 8	7	6 4	3	2 0
Reserved	TM_LINK_SELECT	Reserved	PD_LINK_SELECT	Reserved	SD_LINK_SELECT	Reserved	RM_LINK_SEL
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 126. EE Link Select Register A for AI_EVENT[2] (EE_LINK_SEL_EV2A[(1)]) Field Descriptions

Bit	Field	Value	Description
31-27	Reserved	0	Reserved
26-24	DB_LINK_SELECT		Link select for DB errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
23	Reserved	0	Reserved
22-20	CD_LINK_SELECT		Link select for CD errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
19	Reserved	0	Reserved
18-16	AG_LINK_SELECT		Link select for AG errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
15	Reserved	0	Reserved

(1) There is a similar register for AI_EVENT[3], EE_LINK_SEL_EV3A.

Table 126. EE Link Select Register A for AI_EVENT[2] (EE_LINK_SEL_EV2A) Field Descriptions (continued)

Bit	Field	Value	Description
14-12	TM_LINK_SELECT		Link select for TM errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
11	Reserved	0	Reserved
10-8	PD_LINK_SELECT		Link select for PD errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
7	Reserved	0	Reserved
6-4	SD_LINK_SELECT		Link select for SD errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
3	Reserved	0	Reserved
2-0	RM_LINK_SEL		Link select for Rx MAC errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved

8.2.14.3 EE Link Select Register B for AI_EVENT[2] (EE_LINK_SEL_EVB2)

The EE link select register B for AI_EVENT[2] (EE_LINK_SEL_EVB2) is shown in Figure 141 and described in Table 127.

Figure 141. EE Link Select Register B for AI_EVENT[2] (EE_LINK_SEL_EVB2)

31	16
Reserved	
R-0	

15	7	6	4	3	2	0
Reserved		PE_LINK_SELECT		Reserved	CI_LINK_SEL	
R-0		R/W-0		R-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

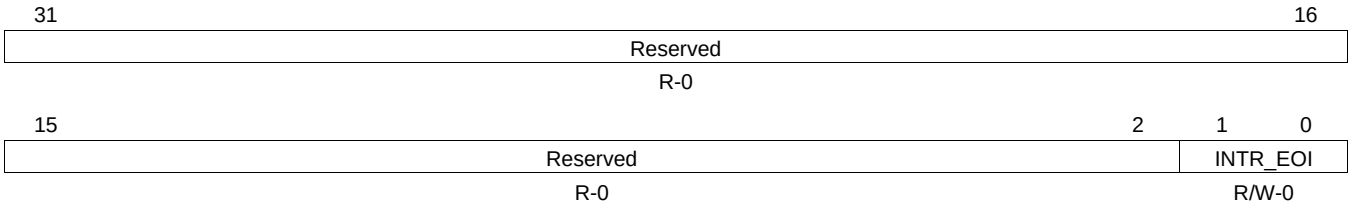
Table 127. EE Link Select Register B for AI_EVENT[2] (EE_LINK_SEL_EVB2) Field Descriptions

Bit	Field	Value	Description
31-7	Reserved	0	Reserved
6-4	PE_LINK_SELECT	0-7h	Link select for PE errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved
3	Reserved	0	Reserved
2-0	CI_LINK_SEL		Link select for CI errors/alarms aggregated to AI_EVENT[2].
		000	Link 0
		001	Link 1
		010	Link 2
		011	Link 3
		100	Link 4
		101	Link 5
		110	Reserved
		111	Reserved

8.2.14.4 EE End of Interrupt (EE_INT_END) Register

The EE end of interrupt (EE_INT_END) register is shown in Figure 142 and described in Table 128.

Figure 142. EE End of Interrupt (EE_INT_END) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 128. EE End of Interrupt (EE_INT_END) Register Field Descriptions

Bit	Field	Value	Description
31-2	Reserved	0	Reserved
1-0	INTR_EOI		Whenever this register is written, the AI_EVENT[3:0] signal addressed by this field is allowed to generate another event if any aggregated bits in the associated interrupt source masked status registers are set to a 1. This field is ignored if the associated AI_EVENTx_MODE bit in Section 8.2.14.1 is set to trigger mode.
		00	AI_EVENT[0]
		01	AI_EVENT[1]
		10	AI_EVENT[2]
		11	AI_EVENT[3]

8.3 Status Registers

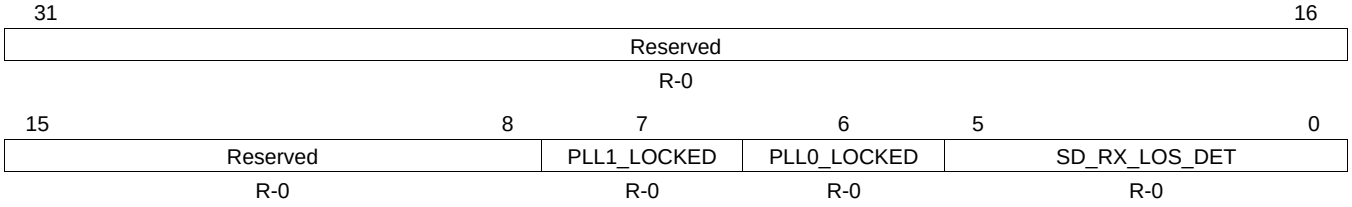
These registers contain fields that reflect the peripheral status.

- Status registers contain status bit fields that are read-only. Bit fields in the status registers are allowed to change due to peripheral actions that are unrelated to any MMR accesses.
- Writes must not affect the status registers or other peripheral logic.
- Status clears are performed using separate command clear registers, as described in the next section.

8.3.1 SERDES Status (SERDES_STS) Register

The SERDES status (SERDES_STS) register is shown in Figure 143 and described in Table 129.

Figure 143. SERDES Status (SERDES_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 129. SERDES Status (SERDES_STS) Register Field Descriptions

Bit	Field	Value	Description
31-8	Reserved	0	Reserved
7	PLL1_LOCKED		Driven high when the PLL1 is locked.
		0	PLL1 is not locked.
		1	PLL1 is locked.
6	PLL0_LOCKED		Driven high when the PLL0 is locked.
		0	PLL0 is not locked.
		1	PLL0 is locked.
5-0	SD_RX_LOS_DET	0-3Fh	Driven high when a loss of signal (electrical idle) condition is detected per link.

8.3.2 Rx MAC Status Registers

8.3.2.1 Rx MAC Link 0 Status (RM_LINK_STSA) Register 0

The Rx MAC link0 status (RM_LINK_STSA) register 0 is shown in Figure 144 and described in Table 130.

Figure 144. Rx MAC Link 0 Status (RM_LINK_STSA) Register 0

31	16
Reserved	
R-0	

15	11	10	9	8	7	4	3	0
Reserved		RM_MSTR_FRAME_BNDY_OUT_OF_RANGE	RM_LOS_DET	RM_LOSS_OF_SIGNAL	Reserved		RM_SYNC_STATUS	
R-0		R-0	R-0	R-0	R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 130. Rx MAC Link 0 Status (RM_LINK_STSA) Register 0 Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	Reserved
10	RM_MSTR_FRAME_BNDY_OUT_OF_RANGE		This error is indicated when a received master frame is detected outside the programmable window: VALID_MSTR_FRAME_WIND. This error is not reported if an extra K28.7 character is detected AND RM_EXTRA_K28P7_ERR_SUPPRESS = 1.
		0	No master frame boundary range error is detected.
		1	Master frame boundary range error is detected.
9	RM_LOS_DET		Detects when the RM_NUM_LOS counter has reached the programmable RM_LOS_DET_THOLD limit.
		0	OBSAI: After a master frame of no 8b10b errors OR when configuration value RM_LOS_DET_THOLD is written. CPRI: After a hyper frame of no 8b10b errors OR when configuration value RM_LOS_DET_THOLD is written.
		1	When RM_NUM_LOS counter has reached RM_LOS_DET_THOLD within a master frame. CPRI: When RM_NUM_LOS counter has reached RM_LOS_DET_THOLD within a hyper frame.
8	RM_LOSS_OF_SIGNAL	1	Active when Rx state machine is in ST0 state, inactive otherwise.
7-4	Reserved	0	Reserved
3-0	RM_SYNC_STATUS	0-8h	Indicates the current status of the Rx state machine.
		0001	ST3
		0010	ST2
		0100	ST1
		1000	ST0

There are six of these registers, one for each link:

RM_LINK0_STSA	RM_LINK2_STSA	RM_LINK4_STSA
RM_LINK1_STSA	RM_LINK3_STSA	RM_LINK5_STSA

8.3.2.2 Rx MAC Link 0 Status (RM_LINK_STSB) Register 1

The Rx MAC link 0 status (RM_LINK_STSB) register 1 is shown in Figure 145 and described in Table 131.

Figure 145. Rx MAC Link 0 Status (RM_LINK_STSB) Register 1

31	20	19	16
RM_RCVD_MSTR_FRAME_OFFSET		RM_NUM_LOS	
R-0		R-0	

15	0
RM_NUM_LOS	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 131. Rx MAC Link 0 Status (RM_LINK_STSB) Register 1 Field Descriptions

Bit	Field	Value	Description
31-20	RM_RCVD_MSTR_FRAME_OFFSET	0-FFFh	Represents the time in VBUS_CLK ticks from after PI_OFFSET that the frame boundary was received. This value saturates at either the programmed RM_RCVD_MSTR_FRAME_WIND threshold or the max value of 4095. This measurement is conducted once per frame. Range: 0 to 4095
19-0	RM_NUM_LOS	0-FFFFh	Represents the number of LOS_DET (8b10b code violation) occurrences in a master frame (OBSAI) or hyper frame (CPRI). This counter saturates and hold its value until it is either cleared by writing the configuration value RM_LOS_DET_THOLD = 0 OR after a master frame (OBSAI) or hyper frame (CPRI) of no 8b10b errors. Range: 0 to 220 – 1

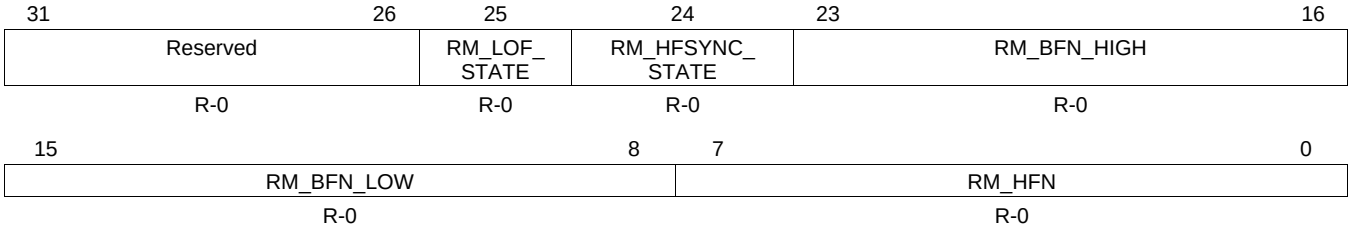
There are six of these registers, one for each link:

RM_LINK0_STSB	RM_LINK2_STSB	RM_LINK4_STSB
RM_LINK1_STSB	RM_LINK3_STSB	RM_LINK5_STSB

8.3.2.3 Rx MAC Link 0 Status (RM_LINK_STSC) Register 2 - CPRI

The Rx MAC link 0 status (RM_LINK_STSC) register 2 - CPRI is shown in Figure 146 and described in Table 132.

Figure 146. Rx MAC Link 0 Status (RM_LINK_STSC) Register 2 - CPRI



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 132. Rx MAC Link 0 Status (RM_LINK_STSC) Register 2 - CPRI Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25	RM_LOF_STATE		Active high status indicates loss of frame when the receiver FSM is in state ST0 or ST1.
		0	Value after reset.
		1	Value if the CPRI mode is enabled.
24	RM_HFSYNC_STATE		Active high status indicates when the receiver FSM is in the HFSYNC state ST3.
		0	The receiver FSM is not in the HFSYNC state ST3.
		1	The receiver FSM is in the HFSYNC state ST3.
23-16	RM_BFN_HIGH	0-FFh	Received Node B Frame number high byte (Z.130.0)
15-8	RM_BFN_LOW	0-FFh	Received Node B Frame number low byte (Z.128.0)
7-0	RM_HFN	0-FFh	Received hyper-frame number (Z.64.0) Range: 0 to 149 basic frames.

There are six of these registers, one for each link:

RM_LINK0_STSC	RM_LINK2_STSC	RM_LINK4_STSC
RM_LINK1_STSC	RM_LINK3_STSC	RM_LINK5_STSC

8.3.2.4 Rx MAC Link 0 Status (RM_LINK_STSD) Register 3 - CPRI

The Rx MAC link 0 status (RM_LINK_STSD) register 3 - CPRI is shown in Figure 147 and described in Table 133.

Figure 147. Rx MAC Link 0 Status (RM_LINK_STSD) Register 3 - CPRI

31	24	23	21	20	19	18	17	16
RM_PNTR_P		Reserved		RM_RCVD_LOF	RM_RCVD_LOS	RM_RCVD_SDI	RM_RCVD_RAI	RM_RCVD_RST
R-0		R-0		R-0	R-0	R-0	R-0	R-0

15	8	7	0
RM_START_UP		RM_VERSION	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 133. Rx MAC Link 0 Status (RM_LINK_STSD) Register 3 - CPRI Field Descriptions

Bit	Field	Value	Description
31-24	RM_PNTR_P	0-FFh	Received Pointer P
23-21	Reserved	0	Reserved
20	RM_RCVD_LOF		LOF status (Z.130.0, b4)
		0	Did not receive LOF.
		1	Received LOF.
19	RM_RCVD_LOS		LOS status (Z.130.0, b3)
		0	Did not receive LOS.
		1	Received LOS.
18	RM_RCVD_SDI		SDI status (Z.130.0, b2)
		0	Did not receive SDI.
		1	Received SDI.
17	RM_RCVD_RAI		RAI status (Z.130.0, b1)
		0	Did not receive RAI.
		1	Received RAI.
16	RM_RCVD_RST		Reset status (Z.130.0, b0)
		0	Did not receive reset.
		1	Received reset.
15-8	RM_START_UP	0-FFh	Received start-up information (Z.66.0)
7-0	RM_VERSION	0-FFh	Received protocol version (Z.2.0)

There are six of these registers, one for each link:

RM_LINK0_STSD	RM_LINK2_STSD	RM_LINK4_STSD
RM_LINK1_STSD	RM_LINK3_STSD	RM_LINK5_STSD

8.3.3 Tx MAC Status (TM_LINK_STS) Registers (Link 0 - Link 5)

The Tx MAC Status (TM_LINK_STS) Registers (Link 0 - Link 5) is shown in Figure 148 and described in Table 134.

Figure 148. Tx MAC Status (TM_LINK_STS) Registers (Link 0 - Link 5)

31	16
Reserved	
R-0	

15 11	10	9	8	7 3	2 0
Reserved	TM_FIFO_OVF	TM_DATA_NOT_ALIGNED	TM_FRAME_NOT_ALIGNED	Reserved	TM_SYNC_STATUS
R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 134. Tx MAC Status (TM_LINK_STS) Registers (Link 0 - Link 5) Field Descriptions

Bit	Field	Value	Description
31-11	Reserved	0	Reserved
10	TM_FIFO_OVF	1	Tx MAC FIFO Overflow flag
9	TM_DATA_NOT_ALIGNED	1	This error is indicated when data to be transmitted is detected as not aligned with frame sync + delta_offset (that is, the Frame boundary to be transmitted is not aligned with frame boundary received internally).
8	TM_FRAME_NOT_ALIGNED	1	This error is indicated when the frame strobe from the frame sync module is not aligned with the frame byte counter.
7-3	Reserved	0	Reserved
2-0	TM_SYNC_STATUS	0-4h	Indicates the current status of the tx state machine.
		001	ST2
		010	ST1
		100	ST0

There are six of these registers, one for each link:

TM_LINK0_STS	TM_LINK2_STS	TM_LINK4_STS
TM_LINK1_STS	TM_LINK3_STS	TM_LINK5_STS

8.3.4 Combiner / Decombiner Status (CD_STS) Register

The combiner / decombiner status (CD_STS) register is shown in Figure 149 and described in Table 135.

Figure 149. Combiner / Decombiner Status (CD_STS) Register

31	22	21	16
Reserved		CD_OUT_FIFO_OVF	
R-0		R-0	

15	14	13	8	7	2	1	0
Reserved		CD_OUT_FIFO_UNF		Reserved		CB_ALIGN_ERR	
R-0		R-0		R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 135. Combiner / Decombiner Status (CD_STS) Register Field Descriptions

Bit	Field	Value	Description
31-22	Reserved	0	Reserved
21-16	CD_OUT_FIFO_OVF	0-3Fh	Active high. CD output FIFO overflow flag per link
15-14	Reserved	0	Reserved
13-8	CD_OUT_FIFO_UNF	0-3Fh	Active high. CD output FIFO underflow flag per link
7-2	Reserved	0	Reserved
1-0	CB_ALIGN_ERR	0-3h	Active high. Indicates frame alignment error for each of the two combiners.

8.3.5 Aggregator Status Registers

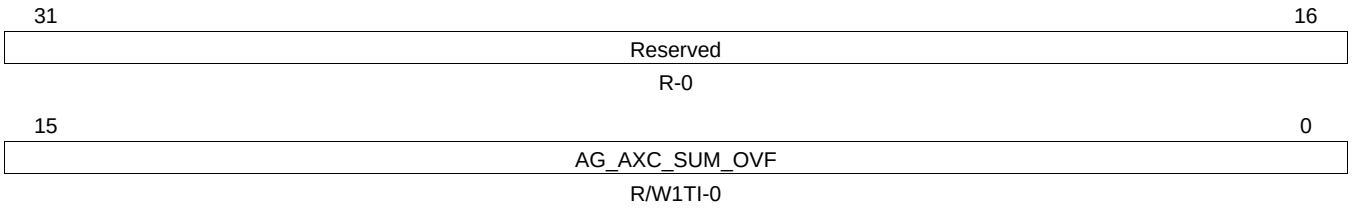
All of non-reserved bits in the aggregator status registers can be inverted by writing a 1 to the bit. This is called a *write 1 to invert* operation. As a result, the traditional *write 1 to clear* access is supported for actual operation, and software can also set a cleared bit for diagnostics.

If the DSP attempts to clear any of the set Aggregator status bits at the same time that the hardware detects a new header error or overflow for that bit, the bit remains set to a 1.

8.3.5.1 Aggregator Link 0 Status (AG_LINK_STS) Register

The aggregator link 0 status (AG_LINK_STS) register is shown in Figure 150 and described in Table 136.

Figure 150. Aggregator Link 0 Status (AG_LINK_STS) Register



LEGEND: R/W1TI = Read/Write 1 to Invert; R = Read only; -*n* = value after reset

Table 136. Aggregator Link 0 Status (AG_LINK_STS) Register Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved
15-0	AG_AXC_SUM_OVF	0-FFFFh	Summation overflow per AxC.

There are six of these registers, one for each link:

AG_LINK0_STS	AG_LINK2_STS	AG_LINK4_STS
AG_LINK1_STS	AG_LINK3_STS	AG_LINK5_STS

8.3.5.2 Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSA) Register 0

The aggregator link 0 header error status (AG_LINK_HDR_ERR_STSA) register 0 is shown in Figure 151 and described in Table 137.

Figure 151. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSA) Register 0

31	21	20	16
Reserved		HDR_ERR	
R-0		R/W1TI-0	

15	0
HDR_ERR	
R/W1TI-0	

LEGEND: R/W1TI = Read/Write 1 to Invert; R = Read only; -*n* = value after reset

Table 137. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSA) Register 0 Field Descriptions

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-0	HDR_ERR		Aggregator header error--one error reported per transmission rule within an 84-slot period as follows:
20		1	Use slot 20 to report an error.
19		1	Use slot 19 to report an error.
18		1	Use slot 18 to report an error.
17		1	Use slot 17 to report an error.
16		1	Use slot 16 to report an error.
15		1	Use slot 15 to report an error.
14		1	Use slot 14 to report an error.
13		1	Use slot 13 to report an error.
12		1	Use slot 12 to report an error.
11		1	Use slot 11 to report an error.
10		1	Use slot 10 to report an error.
9		1	Use slot 9 to report an error.
8		1	Use slot 8 to report an error.
7		1	Use slot 7 to report an error.
6		1	Use slot 6 to report an error.
5		1	Use slot 5 to report an error.
4		1	Use slot 4 to report an error.
3		1	Use slot 3 to report an error.
2		1	Use slot 2 to report an error.
1		1	Use slot 1 to report an error.
0		1	Use slot 0 to report an error.

There are six of these registers, one for each link:

AG_LINK0_HDR_ERR_STSA	AG_LINK2_HDR_ERR_STSA	AG_LINK4_HDR_ERR_STSA
AG_LINK1_HDR_ERR_STSA	AG_LINK3_HDR_ERR_STSA	AG_LINK5_HDR_ERR_STSA

8.3.5.3 Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSB) Register 1

The aggregator link 0 header error status (AG_LINK_HDR_ERR_STSB) register 1 is shown in Figure 152 and described in Table 138.

Figure 152. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSB) Register 1

31	21	20	16
Reserved		HDR_ERR	
R-0		R/W1TI-0	

15	0
HDR_ERR	
R/W1TI-0	

LEGEND: R/W1TI = Read/Write 1 to Invert; R = Read only; -*n* = value after reset

Table 138. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSB) Register 1 Field Descriptions

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-0	HDR_ERR		Aggregator header error--one error reported per transmission rule within an 84-slot period as follows:
20		1	Use slot 41 to report an error.
19		1	Use slot 40 to report an error.
18		1	Use slot 29 to report an error.
17		1	Use slot 38 to report an error.
16		1	Use slot 36 to report an error.
15		1	Use slot 36 to report an error.
14		1	Use slot 35 to report an error.
13		1	Use slot 34 to report an error.
12		1	Use slot 33 to report an error.
11		1	Use slot 32 to report an error.
10		1	Use slot 31 to report an error.
9		1	Use slot 30 to report an error.
8		1	Use slot 29 to report an error.
7		1	Use slot 28 to report an error.
6		1	Use slot 27 to report an error.
5		1	Use slot 26 to report an error.
4		1	Use slot 25 to report an error.
3		1	Use slot 24 to report an error.
2		1	Use slot 23 to report an error.
1		1	Use slot 22 to report an error.
0		1	Use slot 21 to report an error.

There are six of these registers, one for each link:

AG_LINK0_HDR_ERR_STSB	AG_LINK2_HDR_ERR_STSB	AG_LINK4_HDR_ERR_STSB
AG_LINK1_HDR_ERR_STSB	AG_LINK3_HDR_ERR_STSB	AG_LINK5_HDR_ERR_STSB

8.3.5.4 Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSC) Register 2

The aggregator link 0 header error status (AG_LINK_HDR_ERR_STSC) register 2 is shown in Figure 153 and described in Table 139.

Figure 153. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSC) Register 2

31	21	20	16
Reserved		HDR_ERR	
R-0		R/W1TI-0	

15	0
HDR_ERR	
R/W1TI-0	

LEGEND: R/W1TI = Read/Write 1 to Invert; R = Read only; -*n* = value after reset

Table 139. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSC) Register 2

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-0	HDR_ERR		Aggregator header error--one error reported per transmission rule within an 84-slot period as follows:
20		1	Use slot 62 to report an error.
19		1	Use slot 61 to report an error.
18		1	Use slot 60 to report an error.
17		1	Use slot 59 to report an error.
16		1	Use slot 58 to report an error.
15		1	Use slot 57 to report an error.
14		1	Use slot 56 to report an error.
13		1	Use slot 55 to report an error.
12		1	Use slot 54 to report an error.
11		1	Use slot 53 to report an error.
10		1	Use slot 52 to report an error.
9		1	Use slot 51 to report an error.
8		1	Use slot 50 to report an error.
7		1	Use slot 49 to report an error.
6		1	Use slot 48 to report an error.
5		1	Use slot 47 to report an error.
4		1	Use slot 46 to report an error.
3		1	Use slot 45 to report an error.
2		1	Use slot 44 to report an error.
1		1	Use slot 43 to report an error.
0		1	Use slot 42 to report an error.

There are six of these registers, one for each link:

AG_LINK0_HDR_ERR_STSC	AG_LINK2_HDR_ERR_STSC	AG_LINK4_HDR_ERR_STSC
AG_LINK1_HDR_ERR_STSC	AG_LINK3_HDR_ERR_STSC	AG_LINK5_HDR_ERR_STSC

8.3.5.5 Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSD) Register 2

The aggregator link 0 header error status (AG_LINK_HDR_ERR_STSD) register 2 is shown in Figure 154 and described in Table 140.

Figure 154. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSD) Register 2

31	21	20	16
Reserved		HDR_ERR	
R-0		R/W1TI-0	

15	0
HDR_ERR	
R/W1TI-0	

LEGEND: R/W1TI = Read/Write 1 to Invert; R = Read only; -*n* = value after reset

Table 140. Aggregator Link 0 Header Error Status (AG_LINK_HDR_ERR_STSD) Register 2

Bit	Field	Value	Description
31-21	Reserved	0	Reserved
20-0	HDR_ERR		Aggregator header error--one error reported per transmission rule within an 84-slot period as follows:
20		1	Use slot 83 to report an error.
19		1	Use slot 82 to report an error.
18		1	Use slot 81 to report an error.
17		1	Use slot 80 to report an error.
16		1	Use slot 79 to report an error.
15		1	Use slot 78 to report an error.
14		1	Use slot 77 to report an error.
13		1	Use slot 76 to report an error.
12		1	Use slot 75 to report an error.
11		1	Use slot 74 to report an error.
10		1	Use slot 73 to report an error.
9		1	Use slot 72 to report an error.
8		1	Use slot 71 to report an error.
7		1	Use slot 70 to report an error.
6		1	Use slot 69 to report an error.
5		1	Use slot 68 to report an error.
4		1	Use slot 67 to report an error.
3		1	Use slot 66 to report an error.
2		1	Use slot 65 to report an error.
1		1	Use slot 64 to report an error.
0		1	Use slot 63 to report an error.

There are six of these registers, one for each link:

AG_LINK0_HDR_ERR_STSD	AG_LINK2_HDR_ERR_STSD	AG_LINK4_HDR_ERR_STSD
AG_LINK1_HDR_ERR_STSD	AG_LINK3_HDR_ERR_STSD	AG_LINK5_HDR_ERR_STSD

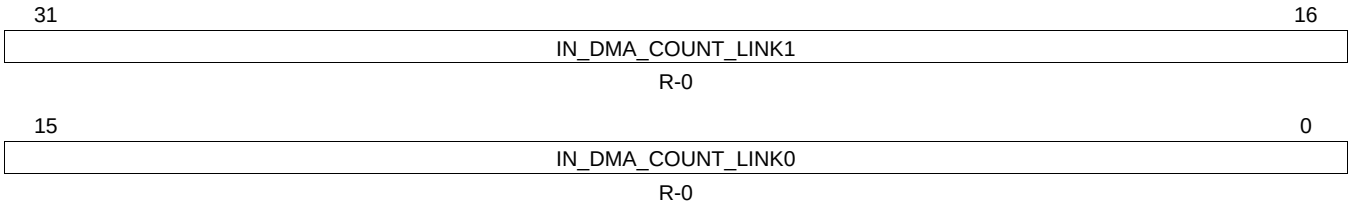
8.3.6 Data Buffer Status Registers

There are several registers that provide the status of the data buffers. The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.1 Data Buffer Inbound DMA Count 0 (DB_IN_DMA_CNT0_STS) Register

The data buffer inbound DMA count 0 (DB_IN_DMA_CNT0_STS) register is shown in Figure 155 and described in Table 141.

Figure 155. Data Buffer Inbound DMA Count 0 (DB_IN_DMA_CNT0_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

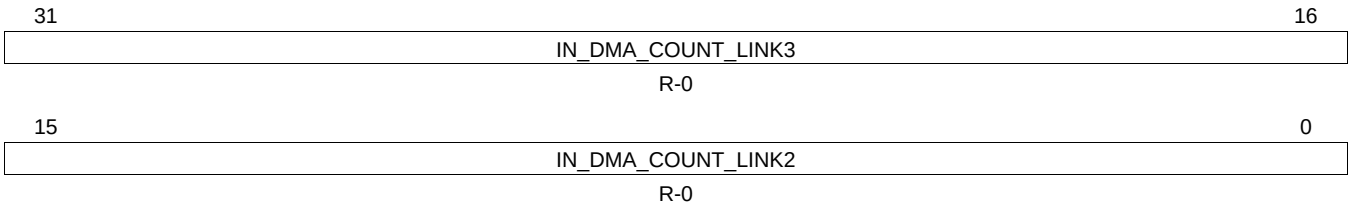
Table 141. Data Buffer Inbound DMA Count 0 (DB_IN_DMA_CNT0_STS) Register Field Descriptions

Bit	Field	Value	Description
31-16	IN_DMA_COUNT_LINK1	0-FFFFh	For Link 1, counts occurrences of DMA inbound done register.
15-0	IN_DMA_COUNT_LINK0	0-FFFFh	For Link 0, counts occurrences of DMA inbound done register.

8.3.6.2 Data Buffer Inbound DMA Count 1 (DB_IN_DMA_CNT1_STS) Register

The data buffer inbound DMA count 1 (DB_IN_DMA_CNT1_STS) register is shown in Figure 156 and described in Table 142.

Figure 156. Data Buffer Inbound DMA Count 1 (DB_IN_DMA_CNT1_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 142. Data Buffer Inbound DMA Count 1 (DB_IN_DMA_CNT1_STS) Register Field Descriptions

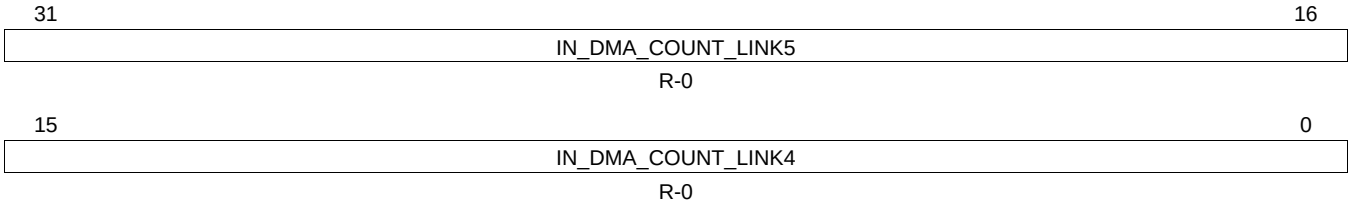
Bit	Field	Value	Description
31-16	IN_DMA_COUNT_LINK3	0-FFFFh	For Link 3, counts occurrences of DMA inbound done register (1).
15-0	IN_DMA_COUNT_LINK2	0-FFFFh	For Link 2, counts occurrences of DMA inbound done register (1).

(1) The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.3 Data Buffer Inbound DMA Count 2 (DB_IN_DMA_CNT2_STS) Register

The data buffer inbound DMA count 2 (DB_IN_DMA_CNT2_STS) register is shown in Figure 157 and described in Table 143.

Figure 157. Data Buffer Inbound DMA Count 2 (DB_IN_DMA_CNT2_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 143. Data Buffer Inbound DMA Count 2 (DB_IN_DMA_CNT2_STS) Register Field Descriptions

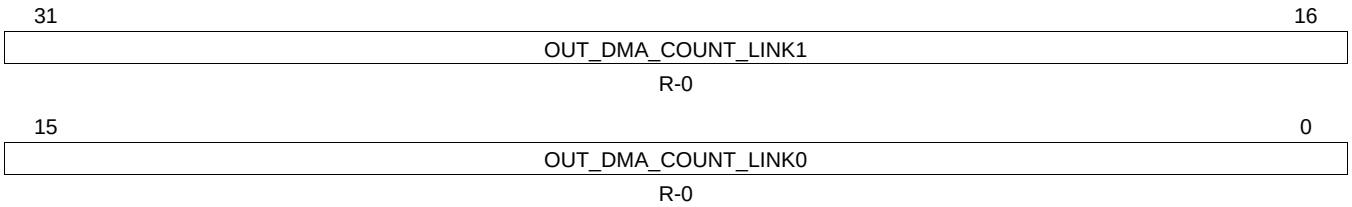
Bit	Field	Value	Description
31-16	IN_DMA_COUNT_LINK5	0-FFFFh	For Link 5, counts occurrences of DMA inbound done register (1).
15-0	IN_DMA_COUNT_LINK4	0-FFFFh	For Link 4, counts occurrences of DMA inbound done register (1).

(1) The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.4 Data Buffer Outbound DMA Count 0 (DB_OUT_DMA_CNT0_STS) Register

The data buffer outbound DMA count 0 (DB_OUT_DMA_CNT0_STS) register is shown in Figure 158 and described in Table 144.

Figure 158. Data Buffer Outbound DMA Count 0 (DB_OUT_DMA_CNT0_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 144. Data Buffer Outbound DMA Count 0 (DB_OUT_DMA_CNT0_STS) Register Field Descriptions

Bit	Field	Value	Description
31-16	OUT_DMA_COUNT_LINK1	0-FFFFh	For Link 1, counts occurrences of DMA outbound done register (1).
15-0	OUT_DMA_COUNT_LINK0	0-FFFFh	For Link 0, counts occurrences of DMA outbound done register (1).

(1) The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.5 Data Buffer Outbound DMA Count 1 (DB_OUT_DMA_CNT1_STS) Register

The data buffer outbound DMA count 1 (DB_OUT_DMA_CNT1_STS) register is shown in Figure 159 and described in Table 145.

Figure 159. Data Buffer Outbound DMA Count 1 (DB_OUT_DMA_CNT1_STS) Register

31	16
OUT_DMA_COUNT_LINK3	
R-0	

15	0
OUT_DMA_COUNT_LINK2	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 145. Data Buffer Outbound DMA Count 1 (DB_OUT_DMA_CNT1_STS) Register Field Descriptions

Bit	Field	Value	Description
31-16	OUT_DMA_COUNT_LINK3	0-FFFFh	For Link 3, counts occurrences of DMA outbound done register (1).
15-0	OUT_DMA_COUNT_LINK2	0-FFFFh	For Link 2, counts occurrences of DMA outbound done register (1).

(1) The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.6 Data Buffer Outbound DMA Count 2 (DB_OUT_DMA_CNT2_STS) Register

The data buffer outbound DMA count 2 (DB_OUT_DMA_CNT2_STS) register is shown in Figure 160 and described in Table 146.

Figure 160. Data Buffer Outbound DMA Count 2 (DB_OUT_DMA_CNT2_STS) Register

31	16
OUT_DMA_COUNT_LINK5	
R-0	

15	0
OUT_DMA_COUNT_LINK4	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 146. Data Buffer Outbound DMA Count 2 (DB_OUT_DMA_CNT2_STS) Register Field Descriptions

Bit	Field	Value	Description
31-16	OUT_DMA_COUNT_LINK5	0-FFFFh	For Link 5, counts occurrences of DMA outbound done register (1).
15-0	OUT_DMA_COUNT_LINK4	0-FFFFh	For Link 4, counts occurrences of DMA outbound done register (1).

(1) The DMA done register is on the DMA/Data bus, not on the configuration bus.

8.3.6.7 Data Buffer Inbound DMA Burst Available (DB_IN_DMA_DEPTH_STS) Register

The data buffer inbound DMA burst available (DB_IN_DMA_DEPTH_STS) register is shown in Figure 161 and described in Table 147.

Figure 161. Data Buffer Inbound DMA Burst Available (DB_IN_DMA_DEPTH_STS) Register

31	23	22	20	19	18	16
Reserved		IN_DMA_DEPTH5		Reserved	IN_DMA_DEPTH4	
R-0		R-0		R-0	R-0	

15	14	12	11	10	8	7	6	4	3	2	0
Reserved	IN_DMA_DEPTH3		Reserved	IN_DMA_DEPTH2		Reserved	IN_DMA_DEPTH1		Reserved	IN_DMA_DEPTH0	
R-0	R-0		R-0	R-0		R-0	R-0		R-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; *-n* = value after reset

Table 147. Data Buffer Inbound DMA Burst Available Register (DB_IN_DMA_DEPTH_STS) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved
22-20	IN_DMA_DEPTH5	0-7h	Link 5 PD-DMA Tracker: Number of DMA bursts written into DB by PD available for read by VBUS. Count 0 to +8. Oflow wrap=0. Uflow wrap=+8. Multiply value of depth by 4x for indication of buffer chip depth. For burst of eight formats, counter increments and decrements by 2s. Under Oflow/Uflow, counter wraps and causes an Error/Alarm error condition.
19	Reserved	0	Reserved
18-16	IN_DMA_DEPTH4	0-7h	Link 4 PD-DMA Tracker(1).
15	Reserved	0	Reserved
14-12	IN_DMA_DEPTH3	0-7h	Link 3 PD-DMA Tracker(1).
11	Reserved	0	Reserved
10-8	IN_DMA_DEPTH2	0-7h	Link 2 PD-DMA Tracker(1).
7	Reserved	0	Reserved
6-4	IN_DMA_DEPTH1	0-7h	Link 1PD-DMA Tracker(1).
3	Reserved	0	Reserved
2-0	IN_DMA_DEPTH0	0-7h	Link 0 PD-DMA Tracker(1).

(1) See the IN_DMA_DEPTH5 bit for details.

8.3.6.8 Data Buffer Outbound DMA Burst Available (DB_OUT_DMA_DEPTH_STS) Register

The data buffer outbound DMA burst available (DB_OUT_DMA_DEPTH_STS) register is shown in Figure 162 and described in Table 148.

Figure 162. Data Buffer Outbound DMA Burst Available (DB_OUT_DMA_DEPTH_STS) Register

31	23	22	20	19	18	16
Reserved		OUT_DMA_DEPTH5		Reserved	OUT_DMA_DEPTH4	
R-0		R-0		R-0	R-0	

15	14	12	11	10	8	7	6	4	3	2	0
Reserved	OUT_DMA_ DEPTH3		Reserved	OUT_DMA_ DEPTH2		Reserved	OUT_DMA_ DEPTH1		Reserved	OUT_DMA_ DEPTH0	
R-0	R-0		R-0	R-0		R-0	R-0		R-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 148. Data Buffer Inbound DMA Burst Available Register (DB_OUT_DMA_DEPTH_STS) Field Descriptions

Bit	Field	Value	Description
31-23	Reserved	0	Reserved
22-20	OUT_DMA_DEPTH5	0-7h	Link 5 PE-DMA Tracker(1): Number of DMA bursts written into DB available for processing by PE. Count 0 to +7. Multiply value of depth by 4x for indication of buffer chip depth. For burst of eight formats, counter will increment and decrement by 2s. Under Oflow/Uflow, counter will wrap and cause Error/Alarm error condition.
19	Reserved	0	Reserved
18-16	OUT_DMA_DEPTH4	0-7h	Link 4 PE-DMA Tracker(2).
15	Reserved	0	Reserved
14-12	OUT_DMA_DEPTH3	0-7h	Link 3 PE-DMA Tracker(2).
11	Reserved	0	Reserved
10-8	OUT_DMA_DEPTH2	0-7h	Link 2 PE-DMA Tracker(2).
7	Reserved	0	Reserved
6-4	OUT_DMA_DEPTH1	0-7h	Link 1 PE-DMA Tracker(2).
3	Reserved	0	Reserved
2-0	OUT_DMA_DEPTH0	0-7h	Link 0 PE-DMA Tracker(2).

(1) PE-FS tracker is not observable, but in well-behaved, normal operation, the PE-DMA and PE-FS trackers have very similar values.
(2) See the OUT_DMA_DEPTH5 bit for details.

8.3.6.9 Data Buffer Outbound Packet-Switched FIFO Status (DB_OUT_PKTSW_STS) Register

The data buffer outbound packet-switched FIFO status (DB_OUT_PKTSW_STS) register is shown in Figure 163 and described in Table 149.

Figure 163. Data Buffer Outbound Packet-Switched FIFO Status (DB_OUT_PKTSW_STS) Register

31	27	26	25	24	16
PKTSW_FIFO_RD_INDEX		Reserved		PKTSW_FIFO_RD_PTR	
R-0		R-0		R-0	

15	9	8	0
Reserved		PKTSW_FIFO_WR_PTR	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 149. Data Buffer Outbound Packet-Switched FIFO Status (DB_OUT_PKTSW_STS) Register Field Descriptions

Bit	Field	Value	Description
31-27	PKTSW_FIFO_RD_INDEX(1)	0-1Fh	FIFO Index 0-to-29 indicating which of 30 FIFOs has the lowest, non-read memory location.
26-25	Reserved	0	Reserved
24-16	PKTSW_FIFO_RD_PTR(1)	0-1FFh	Current state of the lowest Read/head pointer, represents lowest non-read memory location of all 30 FIFOs.
15-9	Reserved	0	Reserved
8-0	PKTSW_FIFO_WR_PTR	0-1FFh	Outbound packet-switched memory, current state of the circular write pointer.

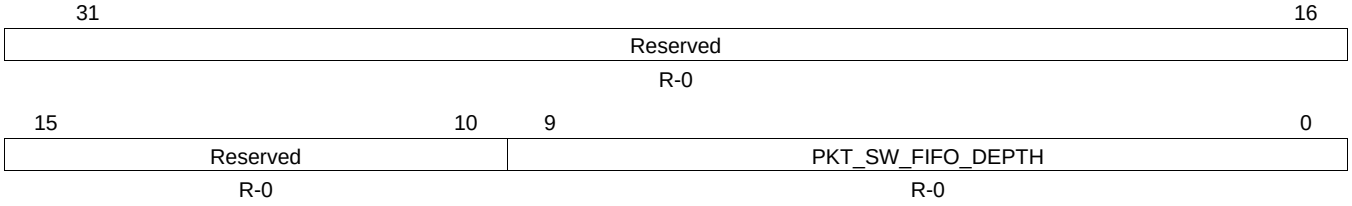
(1) This field requires 32 clock cycles to evaluate. Reading this value while the FIFOs are operating can result in somewhat stale values.

8.3.6.10 Data Buffer Outbound Packet-Switched FIFO Depth (DB_OUT_PKTSW_DEPTH_STS) Register

The flip-flops that comprise the data buffer outbound packet-switched FIFO depth (DB_OUT_PKTSW_DEPTH_STS) register are reset flops and will reset in the 0x0000 state. The depth evaluates to a value of 0x200 32 clock-cycles after reset.

The DB_OUT_PKTSW_DEPTH_STS register is shown in Figure 164 and described in Table 150.

Figure 164. Data Buffer Outbound Packet-Switched FIFO Depth (DB_OUT_PKTSW_DEPTH_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 150. Data Buffer Outbound Packet-Switched FIFO Depth (DB_OUT_PKTSW_DEPTH_STS) Register Field Descriptions

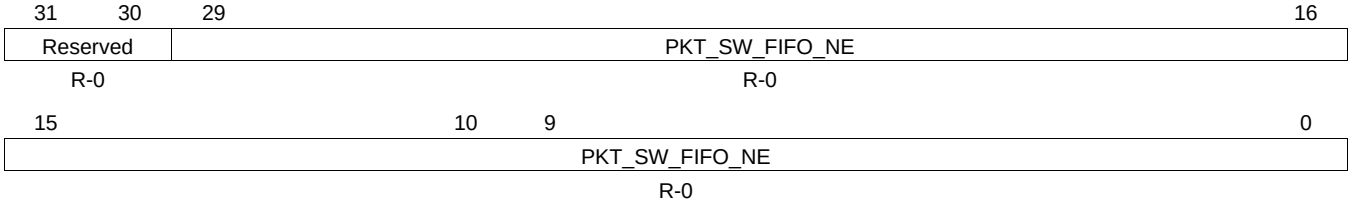
Bit	Field	Value	Description
31-10	Reserved	0	Reserved
9-0	PKT_SW_FIFO_DEPTH (1)	0-200h	Indicates depth of Packet-switched Memory currently available for writes.

(1) This field requires 32 clock cycles to evaluate. Reading this value while the FIFOs are operating can result in somewhat stale values.

8.3.6.11 Data Buffer Outbound Packet-Switched FIFO Not Empty (DB_OUT_PKTSW_NE_STS) Register

The data buffer outbound packet-switched FIFO not empty (DB_OUT_PKTSW_NE_STS) register is shown in Figure 165 and described in Table 151.

Figure 165. Data Buffer Outbound Packet-Switched FIFO Not Empty (DB_OUT_PKTSW_NE_STS) Register



LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 151. Data Buffer Outbound Packet-Switched FIFO Not Empty (DB_OUT_PKTSW_NE_STS) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-0	PKT_SW_FIFO_NE	0-3FFFFFFh	Outbound packet-switched FIFO 0-29, 1'b1 indicates FIFO is currently not empty.

8.3.6.12 Data Buffer Trace Buffer Status (DB_DATA_TRACE_STS) Register

The data buffer trace buffer status (DB_DATA_TRACE_STS) register is shown in Figure 166 and described in Table 152.

Figure 166. Data Buffer Trace Buffer Status (DB_DATA_TRACE_STS) Register

31	30	29	16
Reserved		CAPTURE_RAM_WR_ADR	
R		R-0	

15	0
CAPTURE_RAM_WR_ADR	
R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 152. Data Buffer Trace Buffer Status (DB_DATA_TRACE_STS) Register Field Descriptions

Bit	Field	Value	Description
31-30	Reserved	0	Reserved
29-0	CAPTURE_RAM_WR_ADR	0-3FFFFFFh	Next write address for the data trace capture RAM. (Indicates end of valid data)

8.3.6.13 Data Buffer Outbound Packet-Switched FIFO Head Pointers #0-14 (DB_OUT_PKTSW_HEAD_STS) Register

The data buffer outbound packet-switched FIFO head pointers #0-14 (DB_OUT_PKTSW_HEAD_STS) register is shown in Figure 167 and described in Table 153.

Figure 167. Data Buffer Outbound Packet-Switched FIFO Head Pointers #0-14 (DB_OUT_PKTSW_HEAD_STS) Register

31	25	24	16
Reserved		FIFO_HEAD_PTR1	
R-0		R-0	

15	9	8	0
Reserved		FIFO_HEAD_PTR0	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 153. Data Buffer Outbound Packet-Switched FIFO Head Pointers #0-14 (DB_OUT_PKTSW_HEAD_STS) Register Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24-16	FIFO_HEAD_PTR1	0-1FFh	Packet-switched FIFO #1, Current state of head pointer. For FIFO {1, 3, 5, 7, … 29}
15-9	Reserved	0	Reserved
8-0	FIFO_HEAD_PTR0	0-1FFh	Packet-switched FIFO #0, Current state of head pointer. For FIFO {0, 2, 4, 6, … 28}

There are 15 of these registers, one for each pointer:

DB_OUT_PKTSW_HEAD0_STS	DB_OUT_PKTSW_HEAD5_STS	DB_OUT_PKTSW_HEAD10_STS
DB_OUT_PKTSW_HEAD1_STS	DB_OUT_PKTSW_HEAD6_STS	DB_OUT_PKTSW_HEAD11_STS
DB_OUT_PKTSW_HEAD2_STS	DB_OUT_PKTSW_HEAD7_STS	DB_OUT_PKTSW_HEAD12_STS
DB_OUT_PKTSW_HEAD3_STS	DB_OUT_PKTSW_HEAD8_STS	DB_OUT_PKTSW_HEAD13_STS
DB_OUT_PKTSW_HEAD4_STS	DB_OUT_PKTSW_HEAD9_STS	DB_OUT_PKTSW_HEAD14_STS

8.3.6.14 Data Buffer Outbound Packet-Switched FIFO Tail Pointers #0-14 (DB_OUT_PKTSW_TAIL_STS) Register

The data buffer outbound packet-switched FIFO tail pointers #0-14 (DB_OUT_PKTSW_TAIL_STS) register is shown in Figure 168 and described in Table 154.

Figure 168. Data Buffer Outbound Packet-Switched FIFO Tail Pointers #0-14 (DB_OUT_PKTSW_TAIL_STS) Register

31	25	24	16
Reserved		FIFO_TAIL_PTR1	
R-0		R-0	

15	9	8	0
Reserved		FIFO_TAIL_PTR0	
R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 154. Data Buffer Outbound Packet-Switched FIFO Tail Pointers #0-14 (DB_OUT_PKTSW_TAIL_STS) Register

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24-16	FIFO_TAIL_PTR1	0-1FFh	Packet-switched FIFO #1, Current state of tail pointer. For FIFO {1, 3, 5, 7, … 29}
15-9	Reserved	0	Reserved
8-0	FIFO_TAIL_PTR0	0-1FFh	Packet-switched FIFO #0, Current state of tail pointer. For FIFO {0, 2, 4, 6, … 28}

There are 15 of these registers, one for each pointer:

DB_OUT_PKTSW_TAIL0_STS	DB_OUT_PKTSW_TAIL5_STS	DB_OUT_PKTSW_TAIL10_STS
DB_OUT_PKTSW_TAIL1_STS	DB_OUT_PKTSW_TAIL6_STS	DB_OUT_PKTSW_TAIL11_STS
DB_OUT_PKTSW_TAIL2_STS	DB_OUT_PKTSW_TAIL7_STS	DB_OUT_PKTSW_TAIL12_STS
DB_OUT_PKTSW_TAIL3_STS	DB_OUT_PKTSW_TAIL8_STS	DB_OUT_PKTSW_TAIL13_STS
DB_OUT_PKTSW_TAIL4_STS	DB_OUT_PKTSW_TAIL9_STS	DB_OUT_PKTSW_TAIL14_STS

8.3.7 Exception Event Handler Status Registers

8.3.7.1 EE AIF Run (EE_AI_RUN) Register

The EE AIF run (EE_AI_RUN) register is shown in Figure 169 and described in Table 155.

Figure 169. EE AIF Run (EE_AI_RUN) Register

31	16
Reserved	
R-0	

15	1	0
Reserved		RUN_STAT
R-0		R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 155. EE AIF Run (EE_AI_RUN) Register Field Descriptions

Bit	Field	Value	Description
31-1	Reserved	0	Reserved
0	RUN_STAT		Indicates status of AIF operation. Cleared when AI_EVENT[0] pulses.
		0	AIF not running normally
		1	AIF running normally

Section 8.3.7.2, Section 8.3.7.3, and Section 8.3.7.4 describe the interrupt source raw status registers. Two registers are required for each link and the errors/alarms are grouped by module within the link. There is also a common register for non-link based errors/alarms. In normal operation these registers are read only and the bits are cleared by writing a 1 to the associated interrupt source masked status register. For diagnostics, the bits can be set by writing a 1.

8.3.7.2 EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_A) Register A

The EE link 0 interrupt source raw status (EE_LINK_IRS_A) register A is shown in Figure 170 and described in Table 156.

Figure 170. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_A) Register A

31	29	28	27	26	25	24
Reserved		PD_CPRI_HFN_ERR	PD_OBSAI_TYPE_ERR	PD_OBSAI_ADR_ERR	PD_FSYNC_OR_K_ERR	PD_TIME_STAMP_ERR
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

23	21	20	19	17	16
Reserved		CI_CPRI_FSYNC_ERR	Reserved		SD_LOS
R-0		R/W-0	R-0		R/W-0

15	14	13	12	11	10	9	8
RM_LOF_STATE	RM_HFNSYNC_STATE	RM_RCVD_LOF	RM_RCVD_LOS	RM_RCVD_SDI	RM_RCVD_RAI	RM_RCVD_RST	RM_K30P7_DET
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
RM_MSTR_FRAME_BNDY_OUT_OF_RANGE	RM_MISSING_FRAME_INDICATOR	RM_MISSING_K28P5	RM_BLOCK_BNDRY_DET	RM_FRAME_BNDRY_DET	RM_8B10B_DECODE_ERROR	RM_RCVD_LOS	RM_SYNC_STATUS_CHANGE
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 156. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_A) Register A Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28	PD_CPRI_HFN_ERR(1)	1	CPRI hyper-frame number different than FSM expected
27	PD_OBSAI_TYPE_ERR(1)	1	OBSAI header type look-up resulted in illegal type
26	PD_OBSAI_ADR_ERR(1)	1	OBSAI header address look-up resulted in illegal address
25	PD_FSYNC_OR_K_ERR(1)	1	Frame sync or K character different than FSM expected
24	PD_TIME_STAMP_ERR(1)	1	OBSAI time stamp error
23-21	Reserved	0	Reserved
20	CI_CPRI_FSYNC_ERR(2)	1	This error occurs when a frame boundary is detected when the CI is not expecting it. (CPRI only)
19	Reserved	0	Reserved
16	SD_LOS(3)	1	Loss of signal condition
15	RM_LOF_STATE(4)	1	CPRI receiver entered State 0 or State 1
14	RM_HFNSYNC_STATE(4)	1	CPRI receiver in FSM State 3
13	RM_RCVD_LOF(4)	1	Received LOF (Z.130.0, b4) (CPRI only)
12	RM_RCVD_LOS(4)	1	Received LOS (Z.130.0, b3) (CPRI only)
11	RM_RCVD_SDI(4)	1	Received SDI (Z.130.0, b2) (CPRI only)
10	RM_RCVD_RAI(4)	1	Received RAI (Z.130.0, b1) (CPRI only)
9	RM_RCVD_RST(4)	1	Received reset (Z.130.0, b0) (CPRI only)
8	RM_K30P7_DET(4)	1	Indicates that a K30.7 character was received (OBSAI only).
7	RM_MSTR_FRAME_BNDY_OUT_OF_RANGE(4)	1	This error is indicated when received Master Frame is detected outside the programmable window valid_mstr_frame_wind (typically 65ns)

(1) Per link error/alarm status bits from Protocol Decoder
(2) Per link error/alarm status bits from CPRI Input Data Format Converter
(3) Per link error/alarm status bits from SERDES
(4) Per link error/alarm status bits from Rx MAC

Table 156. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_A) Register A Field Descriptions (continued)

Bit	Field	Value	Description
6	RM_MISSING_FRAME_INDICATOR(4)	1	Indicates that a K28.7 (OBSAI) or a K28.5 (CPRI) character was missing and assumes receiver is in state ST3
5	RM_MISSING_K28P5(4)	1	Indicates that a K28.5 character was missing (OBSAI only and assumes receiver is in frame sync state ST3.
4	RM_BLOCK_BNDRY_DET(4)	1	Block boundary (OBSAI) is detected or at a hyper-frame boundary (CPRI)
3	RM_FRAME_BNDRY_DET(4)	1	Master frame boundary (OBSAI) is detected or at a hyper-frame boundary (CPRI) that delimits a UMTS frame
2	RM_8B10B_DECODE_ERROR(4)	1	8b10b error has occurred.
1	RM_RCVD_LOS(4)	1	LOS_THOLD is reached
0	RM_SYNC_STATUS_CHANGE (4)	1	Indicates the status of the rx state machine.

There are six of these registers, one for each link:

EE_LINK0_IRS_A	EE_LINK2_IRS_A	EE_LINK4_IRS_A
EE_LINK1_IRS_A	EE_LINK3_IRS_A	EE_LINK5_IRS_A

8.3.7.3 EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_B) Register B

The EE link0 interrupt source raw status (EE_LINK_IRS_B) register B is shown in Figure 171 and described in Table 157.

Figure 171. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_B) Register B

31	25	24
Reserved		PE_CWORD_K_ERR
R-0		R/W-0

23	21	20	19	18	17	16
Reserved		DB_CIRSW_OUT_DMA_UFLOW	DB_CIRSW_IN_DMA_UFLOW	DB_CIRSW_OUT_BURST_OFLOW	DB_CIRSW_OUT_DMA_OFLOW	DB_CIRSW_IN_DMA_OFLOW
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
Reserved		CD_OUT_FIFO_UNF	CD_OUT_FIFO_OVF	Reserved	AG_LINK_SUM_OVF	AG_FRM_ALIGN_ERR	AG_LINK_HDR_ERR
R-0		R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

7	4	3	2	1	0
Reserved		TM_FIFO_OVF	TM_DATA_NOT_ALIGNED	TM_FRAME_NOT_ALIGNED	TM_SYNC_STATUS_CHANGE
R-0		R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 157. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_B) Register B

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24	PE_CWORD_K_ERR(1)	1	This error occurs when the CPRI control word from the DSP does not match the k28.5 character expected by the PE.
23-21	Reserved	0	Reserved

(1) Per link error/alarm status bits from Protocol Encoder

Table 157. EE Link 0 Interrupt Source Raw Status (EE_LINK_IRS_B) Register B (continued)

Bit	Field	Value	Description
20	DB_CIRSW_OUT_DMA_UFLOW(2)	1	Circuit-switch RAM Outbound DMA underflow
19	DB_CIRSW_IN_DMA_UFLOW(2)	1	Circuit-switch RAM Inbound DMA underflow
18	DB_CIRSW_OUT_BURST_OFLOW(2)	1	Circuit-switch RAM Outbound RAM burst strobe overflow
17	DB_CIRSW_OUT_DMA_OFLOW(2)	1	Circuit-switch RAM Outbound DMA overflow
16	DB_CIRSW_IN_DMA_OFLOW(2)	1	Circuit-switch RAM Inbound DMA overflow
15-14	Reserved	0	Reserved
13	CD_OUT_FIFO_UNF(3)	1	Output FIFO underflow
12	CD_OUT_FIFO_OVF(3)	1	Output FIFO overflow
11	Reserved	0	Reserved
10	AG_LINK_SUM_OVF(4)	1	Summation overflow on any AxC
9	AG_FRM_ALIGN_ERR(4)	1	Frame alignment error between PE and CD
8	AG_LINK_HDR_ERR(4)	1	Link header error on any message slot
7-4	Reserved	0	Reserved
3	TM_FIFO_OVF(5)	1	Tx MAC FIFO Overflow flag
2	TM_DATA_NOT_ALIGNED(5)	1	This error is indicated when data to be transmitted is detected as not aligned with Master Frame + delta_offset (i.e. Frame boundary to be transmitted is not aligned with frame boundary received internally)
1	TM_FRAME_NOT_ALIGNED(5)	1	This error is indicated when the frame strobe from the frame sync module is not aligned with Master Frame + delta_offset.
0	TM_SYNC_STATUS_CHANGE(5)	1	Indicates a change of tx state machine state

(2) Per link error/alarm status bits from Data Buffer RAM
(3) Per link error/alarm status bits from Combiner/Decombiner
(4) Per link error/alarm status bits from Aggregator
(5) Per link error/alarm status bits from Tx MAC

There are six of these registers, one for each link:

EE_LINK0_IRS_B	EE_LINK2_IRS_B	EE_LINK4_IRS_B
EE_LINK1_IRS_B	EE_LINK3_IRS_B	EE_LINK5_IRS_B

8.3.7.4 EE Common Interrupt Source Raw Status (EE_COMMON_IRS) Register

The EE common interrupt source raw status (EE_COMMON_IRS) register is shown in Figure 172 and described in Table 158.

Figure 172. EE Common Interrupt Source Raw Status (EE_COMMON_IRS) Register

31						26	25	24
Reserved							DB_PKTSW_OUT_WR_ERR	DB_CAPT_RAM_VBUS_ERROR
R-0							R/W-0	R/W-0
23	22	21	20	19	18	17	16	
DB_CIRSW_OUT_RD_DEBUG	DB_CIRSW_IN_WR_DEBUG	DB_PDTSW_OUT_RD_DEBUG	DB_PDTSW_IN_WR_DEBUG	DB_PDTSW_OUT_FIFO_OFLOW	DB_PDTSW_OUT_MEM_LEAK	DB_PDTSW_OUT_RAM_WR	DB_PDTSW_IN_RAM_RD	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
15	14	13	12	11	10	9	8	
DB_PKTSW_IN_FIFO_UFLOW3	DB_PKTSW_IN_FIFO_UFLOW2	DB_PKTSW_IN_FIFO_UFLOW1	DB_PKTSW_IN_FIFO_UFLOW0	DB_PKTSW_IN_FIFO_OFLOW3	DB_PKTSW_IN_FIFO_OFLOW2	DB_PKTSW_IN_FIFO_OFLOW1	DB_PKTSW_IN_FIFO_OFLOW0	
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
7						2	1	0
Reserved							CD1_ALIGN_ERR	CD0_ALIGN_ERR
R-0							R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 158. EE Common Interrupt Source Raw Status (EE_COMMON_IRS[(1)]) Register Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25	DB_PKTSW_OUT_WR_ERR	1	Packet-switched outbound RAM/FIFO illegal write
24	DB_CAPT_RAM_VBUS_ERROR	1	Capture buffer, DMA reading wrong/wr half of RAM
23	DB_CIRSW_OUT_RD_DEBUG	1	Circuit-switched outbound RAM VBUS read (okay for offline debug and delayed streams)
22	DB_CIRSW_IN_WR_DEBUG	1	Circuit-switched inbound RAM VBUS write (okay for offline debug)
21	DB_PDTSW_OUT_RD_DEBUG	1	Packet-switched outbound RAM VBUS read (okay for offline debug)
20	DB_PDTSW_IN_WR_DEBUG	1	Packet-switched inbound RAM VBUS write (okay for offline debug)
19	DB_PDTSW_OUT_FIFO_OFLOW	1	Packet-switched outbound RAM/FIFO overflow write pointer wrap into read
18	DB_PDTSW_OUT_MEM_LEAK	1	Packet-switched outbound RAM/FIFO linked list error
17	DB_PDTSW_OUT_RAM_WR	1	Packet-switched outbound RAM write (OBSAI only; okay for CPRI)
16	DB_PDTSW_IN_RAM_RD	1	Packet-switched inbound RAM read (OBSAI only; okay for CPRI)
15	DB_PKTSW_IN_FIFO_UFLOW3	1	Packet-switched error FIFO DMA underflow
14	DB_PKTSW_IN_FIFO_UFLOW2	1	Packet-switched inbound RAM/FIFO #2 DMA underflow
13	DB_PKTSW_IN_FIFO_UFLOW1	1	Packet-switched inbound RAM/FIFO #1 DMA underflow
12	DB_PKTSW_IN_FIFO_UFLOW0	1	Packet-switched inbound RAM/FIFO #0 DMA underflow
11	DB_PKTSW_IN_FIFO_OFLOW3	1	Packet-switched error FIFO DMA overflow
10	DB_PKTSW_IN_FIFO_OFLOW2	1	Packet-switched inbound RAM/FIFO #2 DMA overflow
9	DB_PKTSW_IN_FIFO_OFLOW1	1	Packet-switched inbound RAM/FIFO #1 DMA overflow
8	DB_PKTSW_IN_FIFO_OFLOW0	1	Packet-switched inbound RAM/FIFO #0 DMA overflow
7-2	Reserved	0	Reserved

(1) The CD1_ALIGN_ERR and CD0_ALIGN_ERR bits are non-link-based error/alarm status bits from the Combiner/Decombiner. All other bits are non-link-based error/alarm status bits from the Data Buffer RAM.

Table 158. EE Common Interrupt Source Raw Status (EE_COMMON_IRS) Register Field Descriptions (continued)

Bit	Field	Value	Description
1	CD1_ALIGN_ERR	1	Rx MAC inputs not aligned to CD #1
0	CD0_ALIGN_ERR	1	Rx MAC inputs not aligned to CD #0

Section 8.3.7.5, Section 8.3.7.6, and Section 8.3.7.7 describe the interrupt source masked status registers. There are separate sets of these registers for each of the AI_EVENT[3:0] signals. There is a set of registers per link for errors/alarms aggregated to AI_EVENT[1:0], while a single set of registers is provided for errors/alarms aggregated to AI_EVENT[3:2].

There is a one-to-one correspondence between each set of these registers and the interrupt source raw status registers described in Section 8.3.7.2, Section 8.3.7.3, and Section 8.3.7.4. The read value for each bit is the logical AND of the corresponding bit in the interrupt source raw status register and the corresponding mask bit, which is controlled by the interrupt status mask set and interrupt status mask clear registers described in Section 8.3.7.5, Section 8.3.7.6, and Section 8.3.7.7. As such, these registers are implemented only as addresses.

Writing a 1 to a masked status bit clears the associated bit in the interrupt source raw status register and therefore, has the effect of clearing the masked status bit as well. If the associated error/alarm is asserted at the same time that the masked status bit is written to a 1, the status bit will not be cleared.

8.3.7.5 EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_A_EV0) Register A

The EE links 0-5 interrupt source masked status for AI_EVENT[0] (EE_LINK_IMS_A_EV0) register A is shown in Figure 173 and described in Table 159.

Figure 173. EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_A_EV0) Register A

31	30	29	28	27	26	25	24
Reserved			PD_CPRI_HFN_ERR	PD_OBSAI_TYPE_ERR	PD_OBSAI_ADR_ERR	PD_FSYNC_OR_K_ERR	PD_TIME_STAMP_ERROR
R-0			R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

23	22	21	20	19	18	17	16
Reserved			CI_CPRI_FSYNC_ERR	Reserved			SD_LOS
R-0			R/W1TC-0	R-0			R/W1TC-0

15	14	13	12	11	10	9	8
RM_LOF_STATE	RM_HFNSYNC_STATE	RM_RCVD_LOF	RM_RCVD_LOS	RM_RCVD_SDI	RM_RCVD_RAI	RM_RCVD_RST	RM_K30P7_DET
R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

7	6	5	4	3	2	1	0
RM_FRAME_BNDY_RNG_ERR	RM_MISSING_FRAME_INDICATOR	RM_MISSING_K28P5	RM_BLOCK_BNDRY_DET	RM_FRAME_BNDRY_DET	RM_8B10B_DECODE_ERROR	RM_LOS_DET	RM_SYNC_STATUS_CHANGE
R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; R/W1TC = Read/Write 1 to clear

Table 159. EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_A_EV0) Register A Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28	PD_CPRI_HFN_ERR(1)	1	CPRI hyper-frame number different than FSM expected
27	PD_OBSAI_TYPE_ERR(1)	1	OBSAI header type look-up resulted in illegal type
26	PD_OBSAI_ADR_ERR(1)	1	OBSAI header address look-up resulted in illegal address
25	PD_FSYNC_OR_K_ERR(1)	1	Frame sync or K character different than FSM expected
24	PD_TIME_STAMP_ERROR(1)	1	OBSAI time stamp error
23-21	Reserved	0	Reserved
20	CI_CPRI_FSYNC_ERR(2)	1	This error occurs when a frame boundary is detected when the CI is not expecting it. (CPRI only)
19-17	Reserved	0	Reserved
16	SD_LOS(3)	1	Loss of signal condition
15	RM_LOF_STATE(4)	1	CPRI receiver entered State 0 or State 1
14	RM_HFNSYNC_STATE(4)	1	CPRI receiver in FSM State 3
13	RM_RCVD_LOF(4)	1	Received LOF (Z.130.0, b4) (CPRI only)
12	RM_RCVD_LOS(4)	1	Received LOS (Z.130.0, b3) (CPRI only)
11	RM_RCVD_SDI(4)	1	Received SDI (Z.130.0, b2) (CPRI only)
10	RM_RCVD_RAI(4)	1	Received RAI (Z.130.0, b1) (CPRI only)
9	RM_RCVD_RST(4)	1	Received reset (Z.130.0, b0) (CPRI only)
8	RM_K30P7_DET(4)	1	Indicates that a K30.7 character was detected (OBSAI only).
7	RM_FRAME_BNDY_RNG_ERR(4)	1	This error is indicated when received Master Frame is detected outside the programmable window valid_mstr_frame_wind (typically 65ns)
6	RM_MISSING_FRAME_INDICATOR(4)	1	Indicates that a K28.7 (OBSAI) or a K28.5 (CPRI) character was missing and assumes receiver is in state ST3
5	RM_MISSING_K28P5(4)	1	Indicates that a K28.5 characters was missing (OBSAI only and assumes receiver is in frame sync state ST3.
4	RM_BLOCK_BNDRY_DET(4)	1	Block boundary (OBSAI) is detected or at a hyper-frame boundary (CPRI)
3	RM_FRAME_BNDRY_DET(4)	1	Master frame boundary (OBSAI) is detected or at a hyper-frame boundary (CPRI) that delimits a UMTS frame
2	RM_8B10B_DECODE_ERROR(4)	1	8b10b error has occurred.
1	RM_LOS_DET(4)	1	LOS_THOLD is reached
0	RM_SYNC_STATUS_CHANGE(4)	1	Indicates the status of the RX state machine.

(1) Per link error/alarm status bits from Protocol Decoder
(2) Per link error/alarm status bits from CPRI Input Data Format Converter
(3) Per link error/alarm status bits from SERDES
(4) Per link error/alarm status bits from Rx MAC

There are six EE links 0-5 interrupt source masked status register A for AI_EVENT[0] registers, one for each link:

EE_LINK0_IMS_A_EV0	EE_LINK2_IMS_A_EV0	EE_LINK4_IMS_A_EV0
EE_LINK1_IMS_A_EV0	EE_LINK3_IMS_A_EV0	EE_LINK5_IMS_A_EV0

There are six EE links 0-5 interrupt source masked status register A for AI_EVENT[1] registers, one for each link:

EE_LINK0_IMS_A_EV1	EE_LINK2_IMS_A_EV1	EE_LINK4_IMS_A_EV1
EE_LINK1_IMS_A_EV1	EE_LINK3_IMS_A_EV1	EE_LINK5_IMS_A_EV1

There is one register for AI_EVENT[2]: EE_LINK_IMS_A_EV2, and one for AI_EVENT[3]: EE_LINK_IMS_A_EV3.

8.3.7.6 EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_B_EV0) Register B

The EE links 0-5 interrupt source masked status register B for AI_EVENT[0] (EE_LINK_IMS_B_EV0) is shown in Figure 174 and described in Table 160.

Figure 174. EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_B_EV0) Register B

31	25	24
Reserved		PE_CWORD_K_ERR
R-0		R/W1TC-0

23	21	20	19	18	17	16
Reserved		CIRSW_OUT_DMA_UFLOW	CIRSW_IN_DMA_UFLOW	CIRSW_OUT_BURST_OFLOW	CIRSW_OUT_DMA_OFLOW	CIRSW_IN_DMA_OFLOW
R-0		R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

15	14	13	12	11	10	9	8
Reserved		CD_FIFO_UNF	CD_FIFO_OVF	Reserved	AG_LINK_SUM_OVF	AG_FRM_ALIGN_ERR	AG_LINK_HDR_ERR
R-0		R/W1TC-0	R/W1TC-0	R-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

7	4	3	2	1	0
Reserved		TM_FIFO_OVF	TM_DATA_NOT_ALIGNED	TM_FRAME__NOT_ALIGNED	TM_SYNC_STATUS_CHANGE
R-0		R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; R/W1TC = Read/Write 1 to clear

Table 160. EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_B_EV0) Register B Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24	PE_CWORD_K_ERR(1)	1	This error occurs when the CPRI control word from the DSP does not match the k28.5 character expected by the PE.
23-21	Reserved	0	Reserved
20	CIRSW_OUT_DMA_UFLOW(2)	1	Circuit-switch RAM outbound DMA underflow
19	CIRSW_IN_DMA_UFLOW(2)	1	Circuit-switch RAM inbound DMA underflow
18	CIRSW_OUT_BURST_OFLOW(2)	1	Circuit-switch RAM outbound RAM burst strobe overflow
17	CIRSW_OUT_DMA_OFLOW(2)	1	Circuit-switch RAM outbound DMA overflow
16	CIRSW_IN_DMA_OFLOW(2)	1	Circuit-switch RAM inbound DMA overflow
15-14	Reserved	0	Reserved
13	CD_FIFO_UNF(3)	1	Output FIFO underflow
12	CD_FIFO_OVF(3)	1	Output FIFO overflow
11	Reserved	0	Reserved
10	AG_LINK_SUM_OVF(4)	1	Summation overflow on any AxC
9	AG_FRM_ALIGN_ERR(4)	1	Frame alignment error between PE and CD

(1) Per link error/alarm status bits from Protocol Encoder
(2) Per link error/alarm status bits from Data Buffer RAM
(3) Per link error/alarm status bits from Combiner/Decombiner
(4) Per link error/alarm status bits from Aggregator

Table 160. EE Links 0-5 Interrupt Source Masked Status for AI_EVENT[0] (EE_LINK_IMS_B_EV0) Register B Field Descriptions (continued)

Bit	Field	Value	Description
8	AG_LINK_HDR_ERR[4]	1	Link header error on any message slot
7-4	Reserved	0	Reserved
3	TM_FIFO_OVF[5]	1	Tx MAC FIFO Overflow flag
2	TM_DATA_NOT_ALIGNED[5]	1	This error is indicated when data to be transmitted is detected as not aligned with Master Frame + delta_offset (i.e. Frame boundary to be transmitted is not aligned with frame boundary received internally)
1	TM_FRAME_NOT_ALIGNED[5]	1	This error is indicated when the frame strobe from the frame sync module is not aligned with master frame + delta_offset.
0	TM_SYNC_STATUS_CHANGE[5]	1	Indicates a change of tx state machine state

(5) Per link error/alarm status bits from Tx MAC

There are six EE links 0-5 interrupt source masked status register B for AI_EVENT[0] registers, one for each link:

EE_LINK0_IMS_B_EV0	EE_LINK2_IMS_B_EV0	EE_LINK4_IMS_B_EV0
EE_LINK1_IMS_B_EV0	EE_LINK3_IMS_B_EV0	EE_LINK5_IMS_B_EV0

There are six EE links 0-5 interrupt source masked status register B for AI_EVENT[1] registers, one for each link:

EE_LINK0_IMS_B_EV1	EE_LINK2_IMS_B_EV1	EE_LINK4_IMS_B_EV1
EE_LINK1_IMS_B_EV1	EE_LINK3_IMS_B_EV1	EE_LINK5_IMS_B_EV1

There is one register for AI_EVENT[2]: EE_LINK_IMS_B_EV2, and one for AI_EVENT[3]: EE_LINK_IMS_B_EV3.

8.3.7.7 EE Common Interrupt Source Masked Status for AI_EVENT[3:0] (EE_COMMON_IMS_EV) Register

The EE common interrupt source masked status register for AI_EVENT[0] (EE_COMMON_IMS_EV0) register is shown in Figure 175 and described in Table 161.

Figure 175. EE Common Interrupt Source Masked Status for AI_EVENT[3:0] (EE_COMMON_IMS_EV) Register

31						26	25	24
Reserved							DB_PKTSW_OUT_FIFO_WR_ERR	DB_CAPT_RAM_VBUS_ERROR
R-0							R/W1TC-0	R/W1TC-0
23	22	21	20	19	18	17	16	
DB_CIRSW_OUT_RD_DEBUG	DB_CIRSW_IN_WR_DEBUG	DB_PKTSW_OUT_RD_DEBUG	DB_PKTSW_IN_WR_DEBUG	DB_PKTSW_OUT_FIFO_OFLOW	DB_PKTSW_OUT_MEM_LEAK	DB_PKTSW_OUT_RAM_WR	DB_PKTSW_IN_RAM_RD	
R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	
15	14	13	12	11	10	9	8	
DB_PKTSW_IN_FIFO_UFLOW3	DB_PKTSW_IN_FIFO_UFLOW2	DB_PKTSW_IN_FIFO_UFLOW1	DB_PKTSW_IN_FIFO_UFLOW0	DB_PKTSW_IN_FIFO_OFLOW3	DB_PKTSW_IN_FIFO_OFLOW2	DB_PKTSW_IN_FIFO_OFLOW1	DB_PKTSW_IN_FIFO_OFLOW0	
R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	R/W1TC-0	
7						2	1	0
Reserved							CD1_ALIGN_ERR	CD0_ALIGN_ERR
R-0							R/W1TC-0	R/W1TC-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset; R/W1TC = Read/Write 1 to clear

Table 161. EE Common Interrupt Source Masked Status for AI_EVENT[3:0](EE_COMMON_IMS_EV[(1)]) Register Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25	DB_PKTSW_OUT_FIFO_WR_ERR	1	Packet-switched outbound RAM/FIFO illegal write
24	DB_CAPT_RAM_VBUS_ERROR	1	Capture buffer, dma reading wrong/wr half of RAM
23	DB_CIRSW_OUT_RD_DEBUG	1	Circuit-switched outbound RAM VBUS read (okay for offline debug and delayed streams)
22	DB_CIRSW_IN_WR_DEBUG	1	Circuit-switched inbound RAM VBUS write (okay for offline debug)
21	DB_PKTSW_OUT_RD_DEBUG	1	Packet-switched outbound RAM VBUS read (okay for offline debug)
20	DB_PKTSW_IN_WR_DEBUG	1	Packet-switched inbound RAM VBUS write (okay for offline debug)
19	DB_PKTSW_OUT_FIFO_OFLOW	1	Packet-switched outbound RAM/FIFO overflow write pointer wrap into read
18	DB_PKTSW_OUT_MEM_LEAK	1	Packet-switched outbound RAM/FIFO linked list error
17	DB_PKTSW_OUT_RAM_WR	1	Packet-switched outbound RAM wr (OBSAI only; okay for CPRI)
16	DB_PKTSW_IN_RAM_RD	1	Packet-switched inbound RAM rd (OBSAI only; okay for CPRI)
15	DB_PKTSW_IN_FIFO_UFLOW3	1	Packet-switched Error FIFO DMA underflow
14	DB_PKTSW_IN_FIFO_UFLOW2	1	Packet-switched inbound RAM/FIFO #2 DMA underflow
13	DB_PKTSW_IN_FIFO_UFLOW1	1	Packet-switched inbound RAM/FIFO #1 DMA underflow
12	DB_PKTSW_IN_FIFO_UFLOW0	1	Packet-switched inbound RAM/FIFO #0 DMA underflow
11	DB_PKTSW_IN_FIFO_OFLOW3	1	Packet-switched Error FIFO DMA overflow
10	DB_PKTSW_IN_FIFO_OFLOW2	1	Packet-switched inbound RAM/FIFO #2 DMA overflow
9	DB_PKTSW_IN_FIFO_OFLOW1	1	Packet-switched inbound RAM/FIFO #1 DMA overflow
8	DB_PKTSW_IN_FIFO_OFLOW0	1	Packet-switched inbound RAM/FIFO #0 DMA overflow

(1) The CD1_ALIGN_ERR and CD0_ALIGN_ERR bits are non-link-based error/alarm status bits from the Combiner/Decombiner. All other bits are non-link-based error/alarm status bits from the Data Buffer RAM.

Table 161. EE Common Interrupt Source Masked Status for AI_EVENT[3:0](EE_COMMON_IMS_EV) Register Field Descriptions (continued)

Bit	Field	Value	Description
7-2	Reserved	0	Reserved
1	CD1_ALIGN_ERR	1	Rx MAC inputs not aligned to CD #1
0	CD0_ALIGN_ERR	1	Rx MAC inputs not aligned to CD #0

There are four of these registers, one for each AI_EVENT signal:

EE_COMMON_IMS_EV0	EE_COMMON_IMS_EV2
EE_COMMON_IMS_EV1	EE_COMMON_IMS_EV3

Section 8.3.7.8, Section 8.3.7.9, and Section 8.3.7.10 describe the interrupt source mask set registers. There are separate sets of these registers for each of the AI_EVENT[3:0] signals. There are per link registers for errors/alarms aggregated to AI_EVENT[1:0], while a single set of registers is provided for errors/alarms aggregated to AI_EVENT[3:2]. There is a one-to-one correspondence between each of these registers and the interrupt source mask status registers described in Section 8.3.7.5, Section 8.3.7.6, and Section 8.3.7.7. An event is aggregated to one of the AI_EVENT[3:0] signals when its corresponding mask bit is set to a 1. The mask bits are set to a 1 by writing a 1, and a read to these registers returns the value of the masks. These registers are implemented only as addresses.

8.3.7.8 EE Link 0 Interrupt Source Mask Set A for AI_EVENT[0] (EE_LINK_MSK_SET_A_EV0) Register

The EE link0 interrupt source mask set A for AI_EVENT[0] (EE_LINK_MSK_SET_A_EV0) register is shown in Figure 176 and described in Table 162.

Figure 176. EE Link 0 Interrupt Source Mask Set A for AI_EVENT[0] (EE_LINK_MSK_SET_A_EV0) Register

31	30	29	28	27	26	25	24
Reserved			PD_CPRI_HFN_ERR_MSK	PD_OBSAI_TYPE_ERR_MSK	PD_OBSAI_ADR_ERR	PD_FSYNC_OR_K_ERR_MSK	PD_TIME_STAMP_ERR_MSK
R-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
23	22	21	20	19	18	17	16
Reserved			CI_CPRI_FSYNC_ERR_MSK	Reserved			SD_LOS_MSK
R-0			R/W-0	R-0			R/W-0
15	14	13	12	11	10	9	8
RM_LOF_STATE_MSK	RM_HFNSYNC_STATE_MSK	RM_RCVD_LOF_MSK	RM_RCVD_LOS_MSK	RM_RCVD_SDI_MSK	RM_RCVD_RAI_MSK	RM_RCVD_RST_MSK	RM_K30P7_DET_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
RM_MSTR_FRAME_BNDY_OUT_OF_RANGE_MSK	RM_MISSING_FRAME_INDICATOR_MSK	RM_MISSING_K28P5_MSK	RM_BLOCK_BNDRY_DET_MSK	RM_FRAME_BNDRY_DET_MSK	RM_8B10B_DECODE_ERR0R_MSK	RM_LOS_DET_MSK	RM_SYNC_STATUS_CHANGE_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 162. EE Link 0 Interrupt Source Mask Set A for AI_EVENT[0] (EE_LINK_MSK_SET_A_EV0[1]) Register Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28	PD_CPRI_HFN_ERR_MSK[2]		Mask for the PD_CPRI_HFN_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
27	PD_OBSAI_TYPE_ERR_MSK[2]		Mask for the PD_OBSAI_TYPE_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
26	PD_OBSAI_ADR_ERR[2]		Mask for the PD_OBSAI_ADR_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
25	PD_FSYNC_OR_K_ERR_MSK[2]		Mask for the PD_FSYNC_OR_K_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
24	PD_TIME_STAMP_ERR_MSK[2]		Mask for the PD_TIME_STAMP_ERR associated bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
23-21	Reserved	0	Reserved
20	CI_CPRI_FSYNC_ERR_MSK[3]		Mask for the CI_CPRI_FSYNC_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
19-17	Reserved	0	Reserved
16	SD_LOS_MSK[4]		Mask for the SD_LOS bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
15	RM_LOF_STATE_MSK[5]		Mask for the RM_LOF_STATE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
14	RM_HFNSYNC_STATE_MSK[5]		Mask for the RM_HFNSYNC_STATE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
13	RM_RCVD_LOF_MSK[5]		Mask for the RM_RCVD_LOF bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
12	RM_RCVD_LOS_MSK[5]		Mask for the RM_RCVD_LOS bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

(1) Reading this register returns the value of the mask bit.
(2) Per link error/alarm status bits the Protocol Decoder
(3) Per link error/alarm status bits from CPRI Input Data Format Converter
(4) Per link error/alarm status bits from SERDES
(5) Per link error/alarm status bits from Rx MAC

Table 162. EE Link 0 Interrupt Source Mask Set A for AI_EVENT[0] (EE_LINK_MSK_SET_A_EV0) Register Field Descriptions (continued)

Bit	Field	Value	Description
11	RM_RCVD_SDI_MSK(5)		Mask for the RM_RCVD_SDI bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
10	RM_RCVD_RAI_MSK(5)		Mask for the RM_RCVD_RAI bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
9	RM_RCVD_RST_MSK(5)		Mask for the RM_RCVD_RST bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
8	RM_K30P7_DET_MSK(5)		Mask for the RM_K30P7_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
7	MSTR_FRAME_BNDY_OUT_OF_RANGE_MSK (5)		Mask for the MSTR_FRAME_BNDY_OUT_OF_RANGE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
6	MISSING_FRAME_INDICATOR_MSK		Mask for the MISSING_FRAME_INDICATOR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
5	RM_MISSING_K28P5_MSK(5)		Mask for the RM_MISSING_K28P5 bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
4	RM_BLOCK_BNDRY_DET_MSK(5)		Mask for the RM_BLOCK_BNDRY_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
3	RM_FRAME_BNDRY_DET_MSK(5)		Mask for the RM_FRAME_BNDRY_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
2	RM_8B10B_DECODE_ERR0R_MSK(5)		Mask for the RM_8B10B_DECODE_ERR0R bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
1	RM_LOS_DET_MSK(5)		Mask for the RM_LOS_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
0	RM_SYNC_STATUS_CHANGE_MSK(5)		Mask for the RM_SYNC_STATUS_CHANGE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

There are six EE links 0-5 interrupt source mask set A for AI_EVENT[0] registers, one for each link:

EE_LINK0_MSK_SET_A_EV0	EE_LINK2_MSK_SET_A_EV0	EE_LINK4_MSK_SET_A_EV0
EE_LINK1_MSK_SET_A_EV0	EE_LINK3_MSK_SET_A_EV0	EE_LINK5_MSK_SET_A_EV0

There are six EE links 0-5 interrupt source mask set A for AI_EVENT[1] registers, one for each link:

EE_LINK0_MSK_SET_A_EV1	EE_LINK2_MSK_SET_A_EV1	EE_LINK4_MSK_SET_A_EV1
EE_LINK1_MSK_SET_A_EV1	EE_LINK3_MSK_SET_A_EV1	EE_LINK5_MSK_SET_A_EV1

There is one register for AI_EVENT[2]: EE_LINK_MASK_SET_A_EV2, and one for AI_EVENT[3]: EE_LINK_MASK_SET_A_EV3.

8.3.7.9 EE Link 0 Interrupt Source Mask Set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0) Register

The EE link0 interrupt source mask set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0) register is shown in Figure 177 and described in Table 163.

Figure 177. EE Link 0 Interrupt Source Mask Set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0) Register

31																24
Reserved																PE_CWORD_K_ERR_MSK
R-0																R/W-0
23								21	20	19	18	17	16			
Reserved									DB_OUT_DMA_UFLOW_MSK	DB_IN_DMA_UFLOW_MSK	DB_OUT_BURST_OFLOW_MSK	DB_OUT_DMA_OFLOW_MSK	DB_IN_DMA_OFLOW_MSK			
R-0									R/W-0	R/W-0	R/W-0	R/W-0	R/W-0			
15				14	13	12	11	10	9	8						
Reserved					CD_FIFO_UNF_MSK	CD_FIFO_OVF_MSK	Reserved	AG_LINK_SUM_OVF_MSK	AG_FRM_ALIGN_ERR_MSK	AG_LINK_HDR_ERR_MSK						
R-0					R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0						
7								4	3	2	1	0				
Reserved									TM_FIFO_OVF_MSK	TM_DATA_NOT_ALIGNED_MSK	TM_FRAME_NOT_ALIGNED_MSK	TM_SYNC_STATUS_CHANGE_MSK				
R-0									R/W-0	R/W-0	R/W-0	R/W-0				

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 163. EE Link 0 Interrupt Source Mask Set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0[(1)]) Register Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24	PE_CWORD_K_ERR_MSK[(2)]		Mask for the PE_CWORD_K_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
23-21	Reserved	0	Reserved

(1) Reading this register returns the value of the mask bit.
(2) Per link error/alarm status bits from Protocol Encoder

Table 163. EE Link 0 Interrupt Source Mask Set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0) Register Field Descriptions (continued)

Bit	Field	Value	Description
20	DB_OUT_DMA_UFLOW_MSK[3]		Mask for the DB_OUT_DMA_UFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
19	DB_IN_DMA_UFLOW_MSK[3]		Mask for the DB_IN_DMA_UFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	
		1	Sets the mask to 1 to enable the associated error/alarm.
18	DB_OUT_BURST_OFLOW_MSK[3]		Mask for the DB_OUT_BURST_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
17	DB_OUT_DMA_OFLOW_MSK[3]		Mask for the DB_OUT_DMA_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
16	DB_IN_DMA_OFLOW_MSK[3]		Mask for the DB_IN_DMA_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Writing this bit to a 1 sets the mask to 1 to enable the associated error/alarm.
15-14	Reserved	0	Reserved
13	CD_OUT_FIFO_UNF_MSK[4]		Mask for the CD_OUT_FIFO_UNF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
12	CD_OUT_FIFO_OVF_MSK[4]		Mask for the CD_OUT_FIFO_OVF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
11	Reserved		Reserved
10	AG_LINK_SUM_OVF_MSK[5]		Mask for the AG_LINK_SUM_OVF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
9	AG_FRM_ALIGN_ERR_MSK[5]		Mask for the AG_FRM_ALIGN_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
8	AG_LINK_HDR_ERR_MSK[5]		Mask for the AG_LINK_HDR_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
7-4	Reserved		Reserved
3	TM_FIFO_OVF_MSK[6]		Mask for the TM_FIFO_OVF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
2	TM_DATA_NOT_ALIGNED_MSK[6]		Mask for the TM_DATA_NOT_ALIGNED bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

(3) Per link error/alarm status bits from Data Buffer RAM
(4) Per link error/alarm status bits from Combiner/Decombiner
(5) Per link error/alarm status bits from Aggregator
(6) Per link error/alarm status bits from Tx MAC

Table 163. EE Link 0 Interrupt Source Mask Set B for AI_EVENT[0] (EE_LINK_MSK_SET_B_EV0) Register Field Descriptions (continued)

Bit	Field	Value	Description
1	TM_FRAME_NOT_ALIGNED_MSK(6)		Mask for the TM_FRAME_NOT_ALIGNED bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
0	TM_SYNC_STATUS_CHANGE_MSK(6)		Mask for the TM_SYNC_STATUS_CHANGE bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

There are six EE links 0-5 interrupt source mask set B for AI_EVENT[0] registers, one for each link:

EE_LINK0_MSK_SET_B_EV0	EE_LINK2_MSK_SET_B_EV0	EE_LINK4_MSK_SET_B_EV0
EE_LINK1_MSK_SET_B_EV0	EE_LINK3_MSK_SET_B_EV0	EE_LINK5_MSK_SET_B_EV0

There are six EE links 0-5 interrupt source mask set B for AI_EVENT[1] registers, one for each link:

EE_LINK0_MSK_SET_B_EV1	EE_LINK2_MSK_SET_B_EV1	EE_LINK4_MSK_SET_B_EV1
EE_LINK1_MSK_SET_B_EV1	EE_LINK3_MSK_SET_B_EV1	EE_LINK5_MSK_SET_B_EV1

There is one register for AI_EVENT[2]: EE_LINK_MASK_SET_B_EV2, and one for AI_EVENT[3]: EE_LINK_MASK_SET_B_EV3.

8.3.7.10 EE Common Interrupt Source Mask Set (EE_COMMON_MSK_SET_EV0) Register for AI_EVENT[0]

The EE common interrupt source mask set (EE_COMMON_MSK_SET_EV0) register for AI_EVENT[0] sets the bits in the EE_COMMON_IMS_EV0 register. EE_COMMON_MSK_SET_EV0 is shown in Figure 178 and described in Table 164.

Figure 178. EE Common Interrupt Source Mask Set (EE_COMMON_MSK_SET_EV0) Register for AI_EVENT[0]

31	26	25	24
Reserved		DB_PKTSW_OUT_FIFO_WR_ERR_MSK	DB_CAPT_RAM_VBUS_ERROR_MSK
R-0		R/W-0	R/W-0

23	22	21	20	19	18	17	16
DB_CIRSW_OUT_RD_DEBUG_MSK	DB_CIRSW_IN_WR_DEBUG_MSK	DB_PKTSW_OUT_RD_DEBUG_MSK	DB_PKTSW_IN_WR_DEBUG_MSK	DB_PKTSW_OUT_FIFO_OFLOW_MSK	DB_PKTSW_OUT_MEM_LEAK_MSK	DB_PKTSW_OUT_RAM_WR_MSK	DB_PKTSW_IN_RAM_RD_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
DB_PKTSW_IN_FIFO_UFLOW3_MSK	DB_PKTSW_IN_FIFO_UFLOW2_MSK	DB_IN_FIFO_UFLOW1_MSK	DB_IN_FIFO_UFLOW0_MSK	DB_IN_FIFO_OFLOW3_MSK	DB_IN_FIFO_OFLOW2_MSK	DB_IN_FIFO_OFLOW1_MSK	DB_IN_FIFO_OFLOW0_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	2	1	0
Reserved		CD1_ALIGN_ERR_MSK	CD0_ALIGN_ERR_MSK
R-0		R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 164. EE Common Interrupt Source Mask Set (EE_COMMON_MSK_SET_EV0[(1)]) Register for AI_EVENT[0] Field Descriptions

Bit	Field	Value	Description
31-26	Reserved		Reserved
25	DB_PKTSW_OUT_FIFO_WR_ERR_MSK		Mask for the DB_PKTSW_OUT_FIFO_WR_ERR bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
24	DB_CAPT_RAM_VBUS_ERROR_MSK		Mask for the DB_CAPT_RAM_VBUS_ERROR bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
23	DB_CIRSW_OUT_RD_DEBUG_MSK		Mask for the DB_CIRSW_OUT_RD_DEBUG bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
22	DB_CIRSW_IN_WR_DEBUG_MSK		Mask for the DB_CIRSW_IN_WR_DEBUG bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

(1) Reading this register returns the value of the mask bit.

Table 164. EE Common Interrupt Source Mask Set (EE_COMMON_MSK_SET_EV0) Register for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
21	DB_PKTSW_OUT_RD_DEBUG_MSK		Mask for the DB_PKTSW_OUT_RD_DEBUG bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
20	DB_PKTSW_IN_WR_DEBUG_MSK		Mask for the DB_PKTSW_IN_WR_DEBUG bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
19	DB_PKTSW_OUT_FIFO_OFLOW_MSK		Mask for the DB_PKTSW_OUT_FIFO_OFLOW bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
18	DB_PKTSW_OUT_MEM_LEAK_MSK		Mask for the DB_PKTSW_OUT_MEM_LEAK bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
17	DB_PKTSW_OUT_RAM_WR_MSK		Mask for the DB_PKTSW_OUT_RAM_WR bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
16	DB_PKTSW_IN_RAM_RD_MSK		Mask for the DB_PKTSW_IN_RAM_RD bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
15	DB_PKTSW_IN_FIFO_UFLOW3_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW3 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
14	DB_PKTSW_IN_FIFO_UFLOW2_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW2 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
13	DB_IN_FIFO_UFLOW1_MSK		Mask for the DB_IN_FIFO_UFLOW1 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
12	DB_IN_FIFO_UFLOW0_MSK		Mask for the DB_IN_FIFO_UFLOW0 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
11	DB_IN_FIFO_OFLOW3_MSK		Mask for the DB_IN_FIFO_OFLOW3 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
10	DB_IN_FIFO_OFLOW2_MSK		Mask for the DB_IN_FIFO_OFLOW2 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

Table 164. EE Common Interrupt Source Mask Set (EE_COMMON_MSK_SET_EV0) Register for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
9	DB_IN_FIFO_OFLOW1_MSK		Mask for the DB_IN_FIFO_OFLOW1 bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
8	DB_IN_FIFO_OFLOW0_MSK		Mask for the DB_IN_FIFO_OFLOW bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
7-2	Reserved	0	Reserved
1	CD1_ALIGN_ERR_MSK(2)		Mask for the CD1_ALIGN_ERR bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.
0	CD0_ALIGN_ERR_MSK(2)		Mask for the CD0_ALIGN_ERR bit in the EE_COMMON_IMS_EV0 register.
		0	Ignored
		1	Sets the mask to 1 to enable the associated error/alarm.

(2) The CD1_ALIGN_ERR_MSK and CD0_ALIGN_ERR_MSK bits are non-link-based error/alarm status bits from the Combiner/Decombiner. All other bits are non-link-based error/alarm status bits from the Data Buffer RAM.

There are four of these registers, one for each AI_EVENT signal:

EE_COMMON_MSK_SET_EV0	EE_COMMON_MSK_SET_EV2
EE_COMMON_MSK_SET_EV1	EE_COMMON_MSK_SET_EV3

Section 8.3.7.11, Section 8.3.7.12, and Section 8.3.7.13 describe the interrupt source mask clear registers. There are separate sets of these registers for each of the AI_EVENT[3:0] signals. There are per link registers for errors/alarms aggregated to AI_EVENT[1:0], while a single set of registers is provided for errors/alarms aggregated to AI_EVENT[3:2]. There is a one-to-one correspondence between each of these registers and the interrupt source mask status registers described in Section 8.3.7.8, Section 8.3.7.9, and Section 8.3.7.10. The mask bits are cleared to a 0 by writing a 1, and a read to these registers returns the value of the masks. These registers are implemented only as addresses.

8.3.7.11 EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_A_EV0) Register A for AI_EVENT[0]

The EE link[0-5] interrupt source mask clear (EE_LINK_MSK_CLR_A_EV0) register A for AI_EVENT[0-3] registers are used to disable errors/alarms for the corresponding event. Reading the EE_LINK_MSK_CLR_A_EV0 register returns the value of the mask bit.

EE_LINK_MSK_CLR_A_EV0 is shown in Figure 179 and described in Table 165.

Figure 179. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_A_EV0) Register A for AI_EVENT[0]

31	29	28	27	26	25	24
Reserved		PD_CPRI_HFN_ERR_MSK	PD_OBSAI_TYPE_ERR_MSK	PD_OBSAI_ADR_ERR_MSK	PD_FSYNC_OR_K_ERR_MSK	PD_TIME_STAMP_ERR_MSK
R-0		R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

23	21	20	19	17	16
Reserved		CI_CPRI_FSYNC_ERR_MSK	Reserved		SD_LOS_MSK
R-0		R/W-0	R-0		R/W-0

15	14	13	12	11	10	9	8
RM_LOF_STATE_MSK	RM_HFNSYNC_STATE_MSK	RM_RCVD_LOF_MSK	RM_RCVD_LOS_MSK	RM_RCVD_SDI_MSK	RM_RCVD_RAI_MSK	RM_RCVD_RST_MSK	RM_K30P7_DET_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
RM_MSTR_FRAME_BNDY_OUT_OF_RANGE_MSK	RM_MISSING_FRAME_INDICATOR_MSK	RM_MISSING_K28P5_MSK	RM_BLOCK_BNDRY_DET_MSK	RM_FRAME_BNDRY_DET_MSK	RM_8B10B_DECODE_ERROR_MSK	RM_LOS_DET_MSK	RM_SYNC_STATUS_CHANGE_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 165. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_A_EV0[(1)]) Register A for AI_EVENT[0] Field Descriptions

Bit	Field	Value	Description
31-29	Reserved	0	Reserved
28	PD_CPRI_HFN_ERR_MSK[(2)]		Mask for the PD_CPRI_HFN_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
27	PD_OBSAI_TYPE_ERR_MSK[(2)]		Mask for the PD_OBSAI_TYPE_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
26	PD_OBSAI_ADR_ERR_MSK[(2)]		Mask for the PD_OBSAI_ADR_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

(1) Reading this register returns the value of the mask bit.
(2) Per link error/alarm status bits from Protocol Decoder

Table 165. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_A_EV0) Register A for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
25	PD_FSYNC_OR_K_ERR_MSK(2)		Mask for the PD_FSYNC_OR_K_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
24	PD_TIME_STAMP_ERR_MSK(2)		Mask for the PD_TIME_STAMP_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
23-21	Reserved	0	Reserved
20	CI_CPRI_FSYNC_ERR_MSK(3)		Mask for the CI_CPRI_FSYNC_ERR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
19-17	Reserved	0	Reserved
16	SD_LOS_MSK(4)		Mask for the SD_LOS bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
15	RM_LOF_STATE_MSK(5)		Mask for the RM_LOF_STATE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
14	RM_HFNSYNC_STATE_MSK(5)		Mask for the RM_HFNSYNC_STATE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
13	RM_RCVD_LOF_MSK(5)		Mask for the RM_RCVD_LOF bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
12	RM_RCVD_LOS_MSK(5)		Mask for the RM_RCVD_LOS bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
11	RM_RCVD_SDI_MSK(5)		Mask for the RM_RCVD_SDI bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
10	RM_RCVD_RAI_MSK(5)		Mask for the RM_RCVD_RAI bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
9	RM_RCVD_RST_MSK(5)		Mask for the RM_RCVD_RST bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

(3) Per link error/alarm status bits from CPRI Input Data Format Converter
(4) Per link error/alarm status bits from SERDES
(5) Per link error/alarm status bits from Rx MAC

Table 165. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_A_EV0) Register A for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
8	RM_K30P7_DET_MSK[5]		Mask for the RM_K30P7_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
7	RM_MSTR_FRAME_BNDY_OUT_OF_RANGE_MSK[5]		Mask for the RM_MSTR_FRAME_BNDY_OUT_OF_RANGE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
6	RM_MISSING_FRAME_INDICATOR_MSK[5]		Mask for the RM_MISSING_FRAME_INDICATOR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
5	RM_MISSING_K28P5_MSK[5]		Mask for the RM_MISSING_K28P5 bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
4	RM_BLOCK_BNDRY_DET_MSK[5]		Mask for the RM_BLOCK_BNDRY_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
3	RM_FRAME_BNDRY_DET_MSK[5]		Mask for the RM_FRAME_BNDRY_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
2	RM_8B10B_DECODE_ERROR_MSK[5]		Mask for the RM_8B10B_DECODE_ERROR bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
1	RM_LOS_DET_MSK[5]		Mask for the RM_LOS_DET bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
0	RM_SYNC_STATUS_CHANGE_MSK[5]		Mask for the RM_SYNC_STATUS_CHANGE bit in the EE_LINK0_ISM_A_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

There are six EE links 0-5 interrupt source mask clear A for AI_EVENT[0] registers, one for each link:

EE_LINK0_MSK_CLR_A_EV0	EE_LINK2_MSK_CLR_A_EV0	EE_LINK4_MSK_CLR_A_EV0
EE_LINK1_MSK_CLR_A_EV0	EE_LINK3_MSK_CLR_A_EV0	EE_LINK5_MSK_CLR_A_EV0

There are six EE links 0-5 interrupt source mask clear A for AI_EVENT[1] registers, one for each link:

EE_LINK0_MSK_CLR_A_EV1	EE_LINK2_MSK_CLR_A_EV1	EE_LINK4_MSK_CLR_A_EV1
EE_LINK1_MSK_CLR_A_EV1	EE_LINK3_MSK_CLR_A_EV1	EE_LINK5_MSK_CLR_A_EV1

There is one register for AI_EVENT[2]: EE_LINK_MASK_CLR_A_EV2, and one for AI_EVENT[3]: EE_LINK_MASK_CLR_A_EV3.

8.3.7.12 EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_B_EV0) Register B for AI_EVENT[0]

The EE link[0-5] interrupt source mask clear (EE_LINK_MSK_CLR_B_EV0) register B for AI_EVENT[0-3] registers are used to disable errors/alarms for the corresponding event. Reading the EE_LINK_MSK_CLR_B_EV0 register returns the value of the mask bit.

EE_LINK_MSK_CLR_B_EV0 is shown in Figure 180 and described in Table 166.

Figure 180. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_B_EV0) Register B for AI_EVENT[0]

31						25	24
Reserved							PE_CWORD_K_ERR_MSK
R-0							R/W-0
23		21	20	19	18	17	16
Reserved			DB_CIRSW_OUT_DMA_UFLOW_MSK	DB_CIRSW_IN_DMA_UFLOW_MSK	DB_CIRSW_OUT_BURST_OFLOW_MSK	DB_CIRSW_OUT_DMA_OFLOW_MSK	DB_CIRSW_IN_DMA_OFLOW_MSK
R-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
15	14	13	12	11	10	9	8
Reserved		UNF_MSK	OVF_MSK	Reserved	LINK_SUM_OVF_MSK	FRM_ALIGN_ERR_MSK	LINK_HDR_ERR_MSK
R-0		R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0
7			4	3	2	1	0
Reserved				LINK_HDR_ERR_MSK	TM_DATA_NOT_ALIGNED_MSK	TM_FRAME_NOT_ALIGNED_MSK	RM_SYNC_STATUS_CHANGE_MSK
R-0				R/W-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 166. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_B_EV0[(1)]) Register B for AI_EVENT[0] Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24	PE_CWORD_K_ERR_MSK[(2)]		Mask for the PE_CWORD_K_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
23-21	DB_CIRSW_IN_DMA_UFLOW_MSK		Mask for the DB_CIRSW_IN_DMA_UFLOW bit in the EE_LINK0_ISM_B_EV0 register
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
20	DB_CIRSW_OUT_DMA_UFLOW_MSK[(3)]		Mask for the DB_CIRSW_OUT_DMA_UFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
19	Reserved	0	Reserved

(1) Reading this register returns the value of the mask bit.
(2) Per link error/alarm status bits from Protocol Encoder
(3) Per link error/alarm status bits from Data Buffer RAM

Table 166. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_B_EV0) Register B for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
18	DB_OUT_BURST_OFLOW_MSK[3]		Mask for the DB_OUT_BURST_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
17	DB_CIRSW_OUT_DMA_OFLOW_MSK[3]		Mask for the DB_CIRSW_OUT_DMA_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
16	DB_CIRSW_IN_DMA_OFLOW_MSK[3]		Mask for the DB_CIRSW_IN_DMA_OFLOW bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
15-14	Reserved	0	Reserved
13	CD_UNF_MSK[4]		Mask for the CD_UNF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
12	CD_OVF_MSK[4]		Mask for the CD_OVF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
11	Reserved	0	Reserved
10	AG_LINK_SUM_OVF_MSK[5]		Mask for the AG_LINK_SUM_OVF bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
9	AG_FRM_ALIGN_ERR_MSK[5]		Mask for the AG_FRM_ALIGN_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
8	AG_LINK_HDR_ERR_MSK[5]		Mask for the AG_LINK_HDR_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
7-4	Reserved	0	Reserved
3	TM_LINK_HDR_ERR_MSK[6]		Mask for the TM_LINK_HDR_ERR bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
2	TM_DATA_NOT_ALIGNED_MSK[6]		Mask for the TM_DATA_NOT_ALIGNED bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
1	TM_FRAME_NOT_ALIGNED_MSK[6]		Mask for the TM_FRAME_NOT_ALIGNED bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

(4) Per link error/alarm status bits from Combiner/Decombiner
(5) Per link error/alarm status bits from Aggregator
(6) Per link error/alarm status bits from Tx MAC

Table 166. EE Link 0 Interrupt Source Mask Clear (EE_LINK_MSK_CLR_B_EV0) Register B for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
0	RM_SYNC_STATUS_CHANGE_MSK(6)		Mask for the RM_SYNC_STATUS_CHANGE bit in the EE_LINK0_ISM_B_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

There are six EE links 0-5 interrupt source mask clear B for AI_EVENT[0] registers, one for each link:

EE_LINK0_MSK_CLR_B_EV0	EE_LINK2_MSK_CLR_B_EV0	EE_LINK4_MSK_CLR_B_EV0
EE_LINK1_MSK_CLR_B_EV0	EE_LINK3_MSK_CLR_B_EV0	EE_LINK5_MSK_CLR_B_EV0

There are six EE links 0-5 interrupt source mask clear B for AI_EVENT[1] registers, one for each link:

EE_LINK0_MSK_CLR_B_EV1	EE_LINK2_MSK_CLR_B_EV1	EE_LINK4_MSK_CLR_B_EV1
EE_LINK1_MSK_CLR_B_EV1	EE_LINK3_MSK_CLR_B_EV1	EE_LINK5_MSK_CLR_B_EV1

There is one register for AI_EVENT[2]: EE_LINK_MASK_CLR_B_EV2, and one for AI_EVENT[3]: EE_LINK_MASK_CLR_B_EV3.

8.3.7.13 EE Common Interrupt Source Mask Clear (EE_COMMON_MSK_CLR_EV0) Reg for AI_EVENT[0]

The EE common interrupt source mask clear (EE_COMMON_MSK_CLR_EV0) reg for AI_EVENT[0] is shown in Figure 181 and described in Table 167.

Figure 181. EE Common Interrupt Source Mask Clear (EE_COMMON_MSK_CLR_EV0) Reg for AI_EVENT[0]

31	30	29	28	27	26	25	24
Reserved						DB_PKTSW_OUT_FIFO_WR_ERR_MSK	DB_CAPT_RAM_VBUS_ERROR_MSK
R-0						R/W-0	R/W-0

23	22	21	20	19	18	17	16
DB_CIRSW_OUT_RD_DEBUG_MSK	DB_CIRSW_IN_WR_DEBUG_MSK	DB_PKTSW_OUT_RD_DEBUG_MSK	DB_PKTSW_IN_WR_DEBUG_MSK	DB_PKTSW_OUT_FIFO_OFLOW_MSK	DB_PKTSW_OUT_MEM_LEAK_MSK	DB_PKTSW_OUT_RAM_WR_MSK	DB_PKTSW_IN_RAM_RD_MSK
R/W-0	R/W-0	R/W-0	R/W09	R/W-0	R/W-0	R/W-0	R/W-0

15	14	13	12	11	10	9	8
DB_PKTSW_IN_FIFO_UFLOW3_MSK	DB_PKTSW_IN_FIFO_UFLOW2_MSK	DB_PKTSW_IN_FIFO_UFLOW1_MSK	DB_PKTSW_IN_FIFO_UFLOW0_MSK	Reserved	DB_PKTSW_IN_FIFO_OFLOW2_MSK	DB_PKTSW_IN_FIFO_OFLOW1_MSK	DB_PKTSW_IN_FIFO_OFLOW0_MSK
R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0

7	6	5	4	3	2	1	0
Reserved						CD1_ALIGN_ERR_MSK	CD0_ALIGN_ERR_MSK
R-0						R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 167. EE Common Interrupt Source Mask Clear (EE_COMMON_MSK_CLR_EV0[1] [2]) Reg for AI_EVENT[0] Field Descriptions

Bit	Field	Value	Description
31-26	Reserved	0	Reserved
25	DB_PKTSW_OUT_FIFO_WR_ERR_MSK		Mask for the DB_PKTSW_OUT_FIFO_WR_ERR bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
24	DB_CAPT_RAM_VBUS_ERROR_MSK		Mask for the DB_CAPT_RAM_VBUS_ERROR bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
23	DB_CIRSW_OUT_RD_DEBUG_MSK		Mask for the DB_CIRSW_OUT_RD_DEBUG bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
22	DB_CIRSW_IN_WR_DEBUG_MSK		Mask for the DB_CIRSW_IN_WR_DEBUG bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
21	DB_KTSW_OUT_RD_DEBUG_MSK		Mask for the DB_KTSW_OUT_RD_DEBUG bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
20	DB_PKTSW_IN_WR_DEBUG_MSK		Mask for the DB_PKTSW_IN_WR_DEBUG bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
19	DB_PKTSW_OUT_FIFO_OFLOW_MSK		Mask for the DB_PKTSW_OUT_FIFO_OFLOW bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
18	DB_PKTSW_OUT_MEM_LEAK_MSK		Mask for the DB_PKTSW_OUT_MEM_LEAK bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
17	DB_PKTSW_OUT_RAM_WR_MSK		Mask for the DB_PKTSW_OUT_RAM_WR bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
16	DB_PKTSW_IN_RAM_RD_MSK		Mask for the DB_PKTSW_IN_RAM_RD bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
15	DB_PKTSW_IN_FIFO_UFLOW3_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW3 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
14	DB_PKTSW_IN_FIFO_UFLOW2_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW2 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

(1) Reading this register returns the value of the mask bit.
(2) The CD1_ALIGN_ERR_MSK and CD0_ALIGN_ERR_MSK bits are non-link-based error/alarm status bits from the Combiner/Decombiner. All other bits are non-link-based error/alarm status bits from the Data Buffer RAM.

Table 167. EE Common Interrupt Source Mask Clear (EE_COMMON_MSK_CLR_EV0) Reg for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
13	DB_PKTSW_IN_FIFO_UFLOW1_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW1 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
12	DB_PKTSW_IN_FIFO_UFLOW0_MSK		Mask for the DB_PKTSW_IN_FIFO_UFLOW0 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
11	Reserved	0	Reserved
10	DB_PKTSW_IN_FIFO_OFLOW2_MSK		Mask for the DB_PKTSW_IN_FIFO_OFLOW2 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
9	DB_PKTSW_IN_FIFO_OFLOW1_MSK		Mask for the DB_PKTSW_IN_FIFO_OFLOW1 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Writing this bit to a 1 clears the mask to 0 to disable the associated error/alarm.
8	DB_PKTSW_IN_FIFO_OFLOW0_MSK		Mask for the DB_PKTSW_IN_FIFO_OFLOW0 bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
7-2	Reserved	0	Reserved
1	CD1_ALIGN_ERR_MSK		Mask for the CD1_ALIGN_ERR bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.
0	CD0_ALIGN_ERR_MSK		Mask for the CD0_ALIGN_ERR bit in the EE_LINK0_COMMON_EV0 register.
		0	Ignored
		1	Clears the mask to 0 to disable the associated error/alarm.

There are four of these registers, one for each AI_EVENT signal:

EE_COMMON_MSK_CLR_EV0 EE_COMMON_MSK_CLR_EV2

EE_COMMON_MSK_CLR_EV1 EE_COMMON_MSK_CLR_EV3

8.3.7.14 EE Interrupt Vector (EE_INT_VECT_EV0-1) Register for AI_EVENT[0-1]

The following tables describe the interrupt vector registers. These are used to direct the DSP to the source of the error/alarm. There are separate registers for each of the AI_EVENT[3:0] signals. Each register has a bit corresponding to each interrupt source masked status register for that event. A set bit indicates that the corresponding interrupt source masked status register has an active error/alarm condition. The bit remains set until all of the active error/alarm conditions are cleared by the DSP.

The EE interrupt vector (EE_INT_VECT_EV0-1) register for AI_EVENT[0-1] is shown in Figure 182 and described in Table 168.

Figure 182. EE Interrupt Vector (EE_INT_VECT_EV0-1) Register for AI_EVENT[0-1]

31	30	29	28	27	26	25	24
Reserved							
R-0							

23	22	21	20	19	18	17	16
Reserved							
R-0							

15	14	13	12	11	10	9	8
Reserved			COMMON_STATUS	LINK5_B_STATUS	LINK5_A_STATUS	LINK4_B_STATUS	LINK4_A_STATUS
R-0			R-0	R-0	R-0	R-0	R-0

7	6	5	4	3	2	1	0
LINK3_B_STATUS	LINK3_A_STATUS	LINK2_B_STATUS	LINK2_A_STATUS	LINK1_B_STATUS	LINK1_A_STATUS	LINK0_B_STATUS	LINK0_A_STATUS
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 168. EE Interrupt Vector (EE_INT_VECT_EV0 [(1)]) Register for AI_EVENT[0] Field Descriptions

Bit	Field	Value	Description
31-13	Reserved	0	Reserved
12	COMMON_STATUS		Status for EE_COMMON_IMS_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
11	LINK5_B_STATUS		Status for EE_LINK5_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
10	LINK5_A_STATUS		Status for EE_LINK5_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
9	LINK4_B_STATUS		Status for EE_LINK4_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
8	LINK4_A_STATUS		Status for EE_LINK4_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
7	LINK3_B_STATUS		Status for EE_LINK3_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register

(1) The EE_INT_VECT_EV1 register is for AI_EVENT[1]. Each bit provides the status for the corresponding bit in the EE_LINK#_IMS_A/B_EV1 register. For instance, LINK2_A_STATUS provides the status of EE_LINK2_IMS_A_EV1.

Table 168. EE Interrupt Vector (EE_INT_VECT_EV0) Register for AI_EVENT[0] Field Descriptions (continued)

Bit	Field	Value	Description
6	LINK3_A_STATUS		Status for EE_LINK3_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
5	LINK2_B_STATUS		Status for EE_LINK2_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
4	LINK2_A_STATUS		Status for EE_LINK2_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
3	LINK1_B_STATUS		Status for EE_LINK1_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
2	LINK1_A_STATUS		Status for EE_LINK1_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
1	LINK0_B_STATUS		Status for EE_LINK0_IMS_B_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
0	LINK0_A_STATUS		Status for EE_LINK0_IMS_A_EV0 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register

8.3.7.15 EE Interrupt Vector (EE_INT_VECT_EV2-3) Register for AI_EVENT[2]

The EE interrupt vector (EE_INT_VECT_EV2-3) register for AI_EVENT[2-3] is shown in Figure 183 and described in Table 169.

Figure 183. EE Interrupt Vector (EE_INT_VECT_EV2-3) Register for AI_EVENT[2-3]

31	16
Reserved	
R-0	

15	3	2	1	0
Reserved		COMMON_STATUS	B_STATUS	A_STATUS
R-0		R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -*n* = value after reset

Table 169. EE Interrupt Vector (EE_INT_VECT_EV2[(1)]) Register for AI_EVENT[2] Field Descriptions

Bit	Field	Value	Description
31-3	Reserved	0	Reserved
2	COMMON_STATUS		Status for EE_COMMON_EV2 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
1	B_STATUS		Status for EE_LINK1_IMS_B_EV2 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register
0	A_STATUS		Status for EE_LINK_IMS_A_EV2 register:
		0	No active error/alarm in register
		1	1 or more active errors/alarms in register

(1) The EE_INT_VECT_EV3 register is for AI_EVENT[3].

8.3.8 VBUSP Configuration Bus Interface Status Registers

8.3.8.1 VC Bus Error (TAIL) Register

The VC bus error (VC_BUSERR) register is shown in Figure 184 and described in Table 170.

The VC_BUSERR register is cleared by writing to it.

Figure 184. VC Bus Error (VC_BUSERR) Register

31	25	24	23	22	20	19	16
Reserved		DIR	Reserved	STATUS		BYTEN	
R-0		R/WTC-0	R-0	R/WTC-0		R/WTC-0	

15	0
ADDR	
R/WTC-0	

LEGEND: R = Read only; WTC = Write to clear

Table 170. VC Bus Error (VC_BUSERR) Register Field Descriptions

Bit	Field	Value	Description
31-25	Reserved	0	Reserved
24	DIR		State of VBUSP dir:
		0	Write
		1	Read
23	Reserved	0	Reserved
22-20	STATUS		VBUSP Error status
		001	Addressing error
		100	Data error
19-16	BYTE_EN	0-Fh	VBUSP byte enables
15-0	ADDR	0-FFFFh	VBUSP configuration bus address that resulted in a bus error. These are VBUS address bits [17:2]

8.3.9 VBUSP DMA Bus Interface Status Registers

8.3.9.1 VBUSP DMA Bus Interface Status (VD_RD_BUSERR) Registers

The VBUSP DMA bus interface status (VD_RD_BUSERR) registers is shown in Figure 185 and described in Table 171.

The VD_RD_BUSERR register is cleared by writing to it.

Figure 185. VBUSP DMA Bus Interface Status (VD_RD_BUSERR) Registers

31	24	23	16
Reserved		ADDR	
R-0		R/WTC-0	

15	0
ADDR	
R/WTC-0	

LEGEND: R = Read only; WTC = Write to clear

Table 171. VBUSP DMA Bus Interface Status (VD_RD_BUSERR) Registers Field Descriptions

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-0	ADDR	0-FFFFFFh	VBUSP DMA read-only bus address that resulted in a bus error. These are VBUS address bits [27:4].

8.3.9.2 VD Bus Error (VD_WR_BUSERR) Register

The VD bus error (VD_WR_BUSERR) register is shown in Figure 186 and described in Table 172.

The VD_WR_BUSERR register is cleared by writing to it.

Figure 186. VD Bus Error (VD_WR_BUSERR) Register

31	24	23	16
Reserved		ADDR	
R-0		R/WTC-0	

15	0
ADDR	
R/WTC-0	

LEGEND: R = Read only; WTC = Write to clear

Table 172. VD Bus Error (VD_WR_BUSERR) Register

Bit	Field	Value	Description
31-24	Reserved	0	Reserved
23-0	ADDR	0-FFFFFFh	VBUSP DMA write-only bus address that resulted in a bus error. These are VBUS address bits [27:4].

9 VBUSP DMA Bus Address Map

The VBUSP DMA address map is shown in Table 173.

Table 173. VBUSP DMA Bus Address Map

VBUSP DMA Address	Access	Description
0xA000_0000 – 0xA07f_FFFF	R/W	Inbound Circuit Switched RAM Link 0-1
0xA080_0000 – 0xA0FF_FFFF	R/W	Inbound Circuit Switched RAM Link 2-3
0xA100_0000 – 0xA17F_FFFF	R/W	Inbound Circuit Switched RAM Link 4-5
0xA180_0000 – 0xA1FF_FFFF	–	Reserved
0xA200_0000 – 0xA27F_FFFF	R/W	Outbound Circuit Switched RAM Link 0-1
0xA280_0000 – 0xA2FF_FFFF	R/W	Outbound Circuit Switched RAM Link 2-3
0xA300_0000 – 0xA37F_FFFF	R/W	Outbound Circuit Switched RAM Link 4-5
0xA380_0000 – 0xA3FF_FFFF	–	Reserved
0xA400_0000 – 0xA4FF_FFFF	R	Inbound Packet-Switched FIFO (OBSAI Access)
0xA500_0000 – 0xA5FF_FFFF	W	Outbound Packet-Switched FIFO (OBSAI Access)
0xA600_0000 – 0xA6FF_FFFF	R/W	Inbound Packet-Switched RAM (CPRI/Debug Access)
0xA700_0000 – 0xA7FF_FFFF	R/W	Outbound Packet-Switched RAM (CPRI/Debug Access)
0xA800_0000 – 0xA8FF_FFFF	W	DMA Completion Registers
0xA900_0000 – 0xA9FF_FFFF	R/W	Data Trace Capture RAM
0xAa00_0000 – 0xAfFF_FFFF	–	Reserved

LEGEND: R/W = Read/Write; R = Read only; W = Write only; -*n* = value after reset

Appendix A Special Cases

A.1 Calculating Delta for a Given Pi (Using the Combining/Decombining Block)

A.1.1 Definitions

Table A-1 provides the definitions that you need to know in order to calculate delta for a given pi when using the combining/decombining block.

Note: DELTA_OFFSET_MIN = PI_OUT_BCLK_DELAY + AI_IN_PIPE_DELAY + CD_PROC_DELAY + AI_OUT_PIPE_DELAY + TM_FIFO_FILL_MIN

Table A-1. Calculating Delta for a Given Pi Definitions

Term	Definition
DELTA_OFFSET_MIN	The minimum value that works for DELTA_OFFSET so that the TM and CD FIFOs will not run dry.
PI_OUT_BCLK_DELAY	The maximum valid delay of the received frame boundary translated from the VBUS_CLK rate into the output BYTE_CLK rate.
AI_IN_PIPE_DELAY	The cumulative, fixed pipeline delay of the CI. It is translated from the VBUS_CLK rate to the output BYTE_CLK rate.
AI_OUT_PIPE_DELAY	The cumulative, fixed pipeline delay of the CI + CD + AG + CO and translated from the VBUS_CLK rate to the output BYTE_CLK rate.
TM_FIFO_FILL_MIN	The minimum number of bytes that you must write to the TM FIFO so that it does not underflow. This value is translated from the INPUT BYTE_CLK rate to the OUTPUT BYTE_CLK rate. The INPUT BYTE_CLK rate is set equal to the OUTPUT BYTE_CLK rate when you only use the PD to transmit data (i.e., you do not use the CD path). You must write a minimum of four bytes to the TM FIFO to prevent underflow.
TM_FIFO_FILL_THOLD	The time it takes to fill the TM FIFO to the programmed TM_FIFO_FULL_THOLD level and to translate from the INPUT BYTE_CLK rate to the OUTPUT BYTE_CLK rate. The value of TM_FIFO_FULL_THOLD is typically set to the maximum level of 28.
CD_PROC_DELAY	The time it takes for input data to the CD to be available at the output of the CD via its FIFOs. This value varies based on the operation that you are performing (redirection, combining, or decombining), input and output byte clock rates, and the number of links processed.
VBUS_CLK_PD	The period when the VBUS clock in the AIF IN_BCLK_PD is equal to the period of the input (i.e., it is received); and, the byte clock in the AIF OUT_BCLK_PD is equal to the period of the output (i.e., transmit) byte clock.

A.1.2 Calculations

Table A-2 lists the calculations used to determine the delta for a given pi using the combining/decombining block.

Table A-2. Determining the Delta for a Given Pi Calculations

Term	Calculations
PI_OUT_BCLK_DELAY	(RM_PI_OFFSET + RM_VALID_MSTR_FRAME_WIND) × VBUS_CLK_PD/OUT_BCLK_PD
AI_OUT_PIPE_DELAY	10 × VBUS_CLK_PD/OUT_CLK_PD (OBSAI RP3)
AI_OUT_PIPE_DELAY	16 × VBUS_CLK_PD/OUT_CLK_PD (CPRI - redirection only)
AI_IN_PIPE_DELAY	1 × VBUS_CLK_PD/OUT_CLK_PD (OBSAI RP3)
AI_IN_PIPE_DELAY	5 × VBUS_CLK_PD/OUT_CLK_PD (CPRI)
TM_FIFO_FILL_MIN	4 × IN_BCLK_PD/OUT_CLK_PD
CD_PROC_DELAY for redirection…	CD_PROC_DELAY = 0
CD_PROC_DELAY for decombining	n × m × IN_BCLK_PD/OUT_CLK_PD ; where… n = output link order number (0 = 1st output link in order, 1 = 2nd output link in order, etc). m = 19 bytes in a message
CD_PROC_DELAY for combining	CD_PROC_DELAY = 0 if ; CB_FS_OFFSET = RM_PI_OFFSET and CB_VALID_DATA_WIND = RM_VALID_MSTR_FRAME_WIND (typically ~65ns)

A.1.3 Examples

Assume the following in the examples below:

RM_PI_OFFSET = 100

RM_VALID_MSTR_FRAME_WIND = 22

VBUS_CLK_PD = 3NS

AI_OUT_PIPE_DELAY = 11 × VBUS_CLK_PD/OUT_CLK_PD

Example A-1. OBSAI RP3 Redirection of a 2x Rate Link

IN_BCLK_PD = OUT_BCLK_PD = 6.5NS PI_OUT_BCLK_DELAY = (100 + 22) × (3ns ÷ 6.5ns) = 56.3 AI_IN_FIXED_DELAY = 1 × (3ns ÷ 6.5ns) = 0.46 AI_OUT_FIXED_DELAY = 10 × (3ns ÷ 6.5ns) = 4.6 TM_FIFO_FILL_MIN = 4 × (6.5ns ÷ 6.5ns) = 4 CD_PROC_DELAY = 0 DELTA_OFFSET_MIN = 56.3 + 0.46 + 4.6 + 4 + 0 = 66 (rounded up)

Example A-2. Decombine 4x Rate Link into 4 - 1x Rate Links

IN_BCLK_PD = 3.25ns OUT_BCLK_PD = 13ns PI_OUT_BCLK_DELAY = (100 + 22) × 3ns/13ns = 28.2 AI_IN_FIXED_DELAY = 1 × (3ns ÷ 6.5ns) = 0.46 AI_OUT_FIXED_DELAY = 10 × (3ns ÷ 13ns) = 2.3 TM_FIFO_FILL_MIN = 4 × (3.25ns ÷13ns) = 1

first output link: CD_PROC_DELAY = 0 DELTA_OFFSET_MIN = 28.2 + 2.3 + 1 + 0 = 32 (rounded up)

second output link: CD_PROC_DELAY = 1 × 19 × (3.25ns ÷13ns) = 4.75 DELTA_OFFSET_MIN = 28.2 + 2.3 + 1 + 4.75 = 37 (rounded up)

third output link: CD_PROC_DELAY = 2 × 19 × (3.25ns ÷13ns) = 9.5 DELTA_OFFSET_MIN = 28.2 + 2.3 + 1 + 9.5 = 42 (rounded up)

fourth output link: CD_PROC_DELAY = 3 × 19 × (3.25ns ÷13ns) = 14.25 DELTA_OFFSET_MIN = 28.2 + 2.3 + 1 + 14.25 = 47 (rounded up)

Note: In the decombining example in Example A-2 a different DELTA_OFFSET value is used for each transmitter. You can also choose to use the maximum calculated DELTA_OFFSET value (the fourth output link value of 47) for each transmitter. Thus, all transmitters will begin transmitting data at approximately the same time.

There are five message groups worth of CD buffering per output link when decombining. You can take advantage of this feature by programming DELTA_OFFSET with a larger value so that two messages per link are available before the transmitters start transmitting.

Programming the DELTA_OFFSET with a larger value guarantees that none of the FIFOs underflow. In Example A-2, this means: CD_PROC_DELAY = (3 + 4) × 19 × 3.25ns/13ns = 33.25 DELTA_OFFSET_MIN = 28.2 + 0.46 + 2.3 + 1 + 33.25 = 67.75 = 66 (rounded-up).

Example A-3. Combining 4 - 1x Rate Links into a 4x Rate Link

IN_BCLK_PD = 13ns OUT_BCLK_PD = 3.25NS PI_OUT_BCLK_DELAY = (100 + 22) × 3ns/3.25ns = 112.6 AI_IN_FIXED_DELAY = 1 × 3NS/6.5NS = 0.46 AI_OUT_FIXED_DELAY = 10 × 3ns/3.25ns = 9.2 TM_FIFO_FILL_MIN = 4 × 13NS/3.25NS = 17.8 CD_PROC_DELAY = 0 DELTA_OFFSET_MIN = 112.6 + 0.46 + 9.2 + 17.8 + 0 = 185.8 = 140

Note: When mixing input line rates for combining, use the slowest rate in the calculations.

In all cases, if the DELTA_OFFSET value is too high relative to the received frame boundary, the CD's output FIFOs will overflow.

A.2 Calculating a Given PE Frame Sync Offset for a Given Delta

There should be at least 10 byte clocks between the PE sync offset and the delta offset.

Calculation: 1x byte clock = 76.8 Mhz = 13.02 NS 1 chip = 32 subchips = 32 × UMTS_CLK_PD = 32 × 8.138 NS = 260.4 NS; therefore, 20 byte clocks = 13.02 NS × 20 = 260.4 NS = 1 chip

Since it is 2.5 chips for 50 1x byte clocks for OBSAI, an offset of 2 is given.

A.3 Minimum Inbound DMA Timing

This section describes the minimum timing between the frame_sync strobe to the Rx MAC and the frame sync system events which are used to transfer received data in the inbound DB buffers.

dma_inbound_cirsw_min = pi_out_bclk_delay + ai_in_pipe_delay + pd_cirsw_capture_min + burst_4_8_delay

dma_inbound_obsai_pktsw_min = pi_out_bclk_delay + ai_in_pipe_delay + pd_pktsw_capture_min + obsai_msg_length

A.3.1 Definitions

Table A-3 provides the definitions that you need to know in order to calculate the minimum inbound DMA timing.

Table A-3. Minimum Inbound DMA Timing Definitions

Term	Definition
dma_inbound_cirsw_min	The earliest that valid data can be DMAed from an inbound link (relative to the rp1_frame_sync signal from the frame sync module). The dma_inbound_obsai_pktsw_min is the earliest that an inbound OBSAI pktsw FIFO will update its not_empty flag and the fifo can be read (time is relative to the rp1_frame_sync signal).

Table A-3. Minimum Inbound DMA Timing Definitions (continued)

Term	Definition
pd_cirsw_capture_min	The worst case latency through the PD data path and write to RAM. (2-to-1 arbitration gives a 1 clock cycle of uncertainty).
pd_pktsw_capture_min	The worst case latency through the PD data path, write FIFO and activation of fifo_ne flag (6-to-1 arbitration gives a 5 clock cycle of uncertainty).
burst_4_8_delay	The time it takes to buffer up four or eight chips of UMTS data. It is expected that the user would wait for a whole burst of chips (4 chips: downlink 8 chips: uplink) for all antenna carriers prior to beginning a DMA transfer. The unit of this is UMTS chips (each chip equals 260.42ns).
obsai_message_length	19 bytes_clk length OBSAI messages. Three bytes of header and sixteen bytes of payload in each OBSAI message. Each byte is received, one per input byte_clk.

A.3.2 Calculations

Table A-4 lists the calculations used to calculate the minimum inbound DMA timing.

Table A-4. Minimum Inbound DMA Timing Calculations

Term	Calculations
pd_cirsw_capture_min	8 × vbus_clk_pd (OBSAI/CPRI RP3)
pd_pktsw_capture_min	10 × vbus_clk_pd (OBSAI)
burst_4_8_delay	4 × chip_clk_pd (OBSAI/CPRI) or 8 × chip_clk_pd
obsai_message_length	9 × in_bclk_pd (OBSAI)

A.3.3 Example

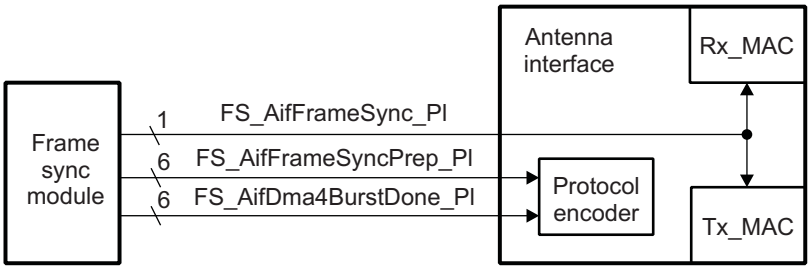
Assume the following:

- OBSAI RP3
- rm_pi_offset = 100
- rm_valid_mstr_frame_wind = 22
- vbus_clk_pd = 3ns
- burst_4_8_delay = 4 umts_chip_clk = 1041.68ns
- link_rate = 2x

Results in these calculations and amounts:

```
in_bclk_pd = 6.5ns
pi_out_bclk_delay = (100 + 22) × 3ns= 366ns
ai_in_fixed_delay =   1 × 3ns= 3ns
obsai_message_length = 19 × 6.5ns = 123.5ns
dma_inbound_cirsw_min = 366 + 3 + 24 + 1041.68 =  1,435ns (rounded up)
dma_inbound_obsai_pktsw_min = 366 + 3 + 30 + 123.5  =  523ns (rounded up)
```

Figure A-1. Sync Events from Frame Sync Module



A.4 Loop Back Test Case Example

The configuration of AIF for a simple loop back test case is given in . The AIF is configured to be in OBSAI mode. The link rate is 1x and both inbound and the outbound are configured to be DL data types. For simplicity sake, the following conditions are assumed:

- Only one link is active.
- Only one packet-switched FIFO is used on both the inbound and outbound sides.
- The address field for the packet-switched slot is 0x0020 (13-bit value) on the outbound side.
- The type field is 0 (control data).

Example A-4. Loop Back Test Case

```
//Declarations
/// Aif link objects
CSL_AifLinkObj AifObj;
// Handle for links
CSL_AifHandle hAifLink;

Void configAif(void)
{
  // Aif pointer to context
  CSL_AifContext AifContext;
  //AIF module specific parameters
  CSL_AifParam  aifParam;
  // CSL status
  CSL_Status status;
  // Ctrl Argument;
  Uint32 ctrlArg;
  /* Setup for links */
  CSL_AifLinkSetup aConfigLink;
  // global config for AIF */
  CSL_AifGlobalSetup gblCfg;
  // Setup for common params for links
  CSL_AifCommonLinkSetup aCommoncfg;
  // Setup for inbound links
  CSL_AifInboundLinkSetup aInboundCfg;
  // Setup for outbound links
  CSL_AifOutboundLinkSetup aOutboundCfg;
  // Setup for Pd for inbound links
  CSL_AifPdSetup aPdCfg;
  // Rx Mac setup for inbound links
  CSL_AifRxMacSetup aRmCfg;
  // Tx Mac setup for outbound links
  CSL_AifTxMacSetup aTmCfg;
  // Protocol encoder setup for outbound links
  CSL_AifPeSetup aPeCfg;
  // Aggegator setup for outbound links
  CSL_AifAggregatorSetup aAgCfg;
  // Serdes setup
  CSL_AifSerdesSetup aSdCfg;
  // Pd setup for all links
  CSL_AifPdCommonSetup pdCommonCfg;
  // Rx MAC setup for all links
  CSL_AifRxMacCommonSetup rmCommonCfg;
  // Serdes common setup
  CSL_AifSerdesCommonSetup sdCommonCfg;
  // Ps FIFO for all links
  CSL_AifInboundFifoSetup psInFifoCfg;
  // Type field LUT
  Uint8 aTypeFieldLut[3] = {0,1,2};
  // Inbound AxC Address field LUT
  Uint16 aAxCAddrLut[4]= {0,1,2,3};

  // Inbound PS FIFO Addr LUT, assuming that the address of the packet slots is
  0x40 after address masking
```

Example A-4. Loop Back Test Case (continued)

```
  Uint16 aPsAddrLut[1][AIF_MAX_SIZE_INBND_PS_ADDR_FIFO] = {0x40};

  // Outbound AxC Addr field LUT
  Uint16 aAxCAddrOutLut [CSL_AIF_MAX_NUM_CS_TRANSMISSION_RULES]=
       {AIF_AXC_ADDR_OUT_LUT_LINK_0};
  // Outbound AxC Mask
  Uint16 aAxCOutMask [CSL_AIF_MAX_NUM_CS_TRANSMISSION_RULES] =
      {AIF_AXC_OUT_MASK_LINK_0};
  // Outbound AxC Compare
  Uint16 aAxCOutCompare [CSL_AIF_MAX_NUM_CS_TRANSMISSION_RULES] =
      {AIF_AXC_OUT_COMP_LINK_0};
  // Outbound AxC Aggregator Ctrl
  Uint8 aAxCAggrCtrlLut [CSL_AIF_MAX_NUM_CS_TRANSMISSION_RULES] =
      {AIF_AXC_OUT_AGGRCTRL_LINK_0};
  // Outbound Ctrl terminal count
  Uint16 aCtrlTCount [CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES] =
      {AIF_CTRL_TCOUNT_LINK_0};
  // Outbound Ctrl compare
  Uint16 aCtrlCmp [CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES] =
      {AIF_CTRL_COMPARE_LINK_0};
  // Outbound Ctrl src select
  CSL_AifOutboundFifoIndex aCtrlSrcSel
  [CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES] =
       {AIF_CTRL_SRC_SEL_LINK_0};
  // Outbound Ctrl data Aggr. Ctrl
  Uint8 aCtrlAggrCtrlLut [CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES] =
      {AIF_CTRL_OUT_AGGRCTRL_LINK_0};
  // counter
  Uint8 i;

 /* Initialize CSL library, this step is required */
 CSL_aifInit(&AifContext);
 aifParam.linkIndex = CSL_AIF_LINK_0;
 /* Open handle for link 0 - for use */
 hAifLink = CSL_aifOpen(&AifObj, CSL_AIF, &aifParam, &status);

 aConfigLink.globalSetup = &gblCfg;
 aConfigLink.commonlinkSetup = &aCommoncfg;
 aConfigLink.inboundlinkSetup = &aInboundCfg;
 aConfigLink.outboundlinkSetup = &aOutboundCfg;

 // populate global config fields
 gblCfg.linkProtocol = CSL_AIF_LINK_PROTOCOL_OBSAI;
 gblCfg.pCdSetup = NULL;
 gblCfg.pInboundPsFifoSetup = &psInFifoCfg;

 //populate common fields
 aCommoncfg.linkIndex = CSL_AIF_LINK_0;
 aCommoncfg.linkRate = CSL_AIF_LINK_RATE_1x;
 aCommoncfg.pSerdesSetup = NULL;//&aSdCfg[0]; // unpopulated for now

 // populate inbound fields for link 0
 aInboundCfg.antDataWidth = CSL_AIF_DATA_WIDTH_16_BIT;
 aInboundCfg.linkDataType = CSL_AIF_LINK_DATA_TYPE_DL;
 aInboundCfg.numActiveAxC = 4;
 aInboundCfg.pRxMacSetup = &aRmCfg;
 aInboundCfg.pPdSetup = &aPdCfg;

 //Populate the outbound fields
 aOutboundCfg.antDataWidth = CSL_AIF_DATA_WIDTH_16_BIT;
 aOutboundCfg.linkDataType = CSL_AIF_LINK_DATA_TYPE_DL;
 aOutboundCfg.numActiveAxC = 4
 aOutboundCfg.pTxMacSetup = &aTmCfg[0];
 aOutboundCfg.pPeSetup = &aPeCfg[0];
```

Example A-4. Loop Back Test Case (continued)

```
aOutboundCfg.pAggrSetup = &aAgCfg[0];
aOutboundCfg.fifoEnBitMask = AIF_DB_OUT_FIFO_ENABLE_MASK_LINK_0;

// PD common cfg
pdCommonCfg.addressMask = 0x3FF;
//Populate the type field LUT of the protocol decoder
pdCommonCfg.numTypeFieldEntries = AIF_NUM_TYPE_FIELD_ENTRIES;
pdCommonCfg.pInboundTypeFieldLut = aTypeFieldLut;

//Program the size of the inbound FIFOs
pdCommonCfg.sizeInboundPsAddrFieldLut[0] = AIF_SIZE_INBND_PS_ADDR_FIFO_0;
pdCommonCfg.sizeInboundPsAddrFieldLut[1] = AIF_SIZE_INBND_PS_ADDR_FIFO_1;
pdCommonCfg.sizeInboundPsAddrFieldLut[2] = AIF_SIZE_INBND_PS_ADDR_FIFO_2;
pdCommonCfg.sizeInboundPsAddrFieldLut[3] = AIF_SIZE_INBND_PS_ADDR_FIFO_3;

//Program the PD address LUT for Packet-switched data
pdCommonCfg.pInboundPsAddrFieldLut[0] = &aPsAddrLut[0];
pdCommonCfg.pInboundPsAddrFieldLut[1] = NULL;
pdCommonCfg.pInboundPsAddrFieldLut[2] = NULL;
pdCommonCfg.pInboundPsAddrFieldLut[3] = NULL;

// populate PD link fields
aPdCfg.bEnablePd = TRUE;
aPdCfg.bCpriCtrlWordCapture = FALSE;
aPdCfg.pPdCommonSetup = &pdCommonCfg;
aPdCfg.sizeInboundAxCAddrFieldLut = 4;
aPdCfg.pInboundAxCAddrFieldLut = &(aAxCAddrLut[0]);

// RM common cfg;
rmCommonCfg.frameSyncT     = 3;
rmCommonCfg.frameUnSyncT   = 3;
rmCommonCfg.syncT          = 3;
rmCommonCfg.unSyncT        = 3;

// populate Rx MAC link fields; Populate the Pi, Valid window values
aRmCfg.bEnableRxMac = TRUE;
aRmCfg.losDetThreshold = 255;
aRmCfg.maxMasterFrameOffset = 200;
aRmCfg.piOffset = 200;
aRmCfg.pRxMacCommonSetup = &rmCommonCfg;
aRmCfg.validMasterFrameOffset = 200;

// populate SD link fields
aSdCfg.bEnableRxAlign = TRUE;
aSdCfg.bEnableRxLos = FALSE;
aSdCfg.pCommonSetup = NULL;
aSdCfg.rxCdrAlgorithm = CSL_AIF_SERDES_RX_CDR_FIRST_ORDER_THRESH_1;
aSdCfg.rxEqualizerConfig = CSL_AIF_SERDES_RX_EQ_ADAPTIVE;
aSdCfg.rxPairPolarity = CSL_AIF_SERDES_TX_PAIR_NORMAL_POLARITY;
aSdCfg.rxTermination = CSL_AIF_SERDES_RX_TERM_COMMON_POINT_VDDT;
aSdCfg.txAmpConfig = CSL_AIF_SERDES_TX_AMP_CONFIG_750;
aSdCfg.txCommonMode = CSL_AIF_SERDES_TX_NORMAL_COMMON_MODE;
aSdCfg.txDeEmphasisConfig = CSL_AIF_SERDES_TX_DE_CONFIG_8;
aSdCfg.txPairPolarity = CSL_AIF_SERDES_TX_PAIR_NORMAL_POLARITY;

// populate Tx MAC link fields
aTmCfg.bEnableTxMac = TRUE;
aTmCfg.bEnableRxLos = FALSE;
aTmCfg.bEnableFrameXmit = TRUE;
aTmCfg.deltaOffset = 50;
aTmCfg.pCpriTxMacSetup = NULL;
aTmCfg.threshTxMacFifo = 8;

// populate PE link fields
```

Example A-4. Loop Back Test Case (continued)

```
aPeCfg.bEnablePe = TRUE;
aPeCfg.cpriCtrlWordMode = CSL_AIF_CPRI_CTRL_WORD_ZEROS;
aPeCfg.obsaiAxCType = AIF_PE_AXC_TYPE_OUTBOUND;

//Populate the 84cnt LUT and the Identity LUT
for (i=0; i < CSL_AIF_MAX_NUM_CS_TRANSMISSION_RULES; i++ )
{
   aPeCfg.aAxCAddressFieldLut[i] = aAxCAddrOutLut[0][i];
   aPeCfg.aAxCMask[i] = aAxCOutMask[0][i];
   aPeCfg.aAxCCompare[i] = aAxCOutCompare[0][i];
   aPeCfg.aAggrCtrlAxCLut[i] = aAxCAggrCtrlLut[0][i];
}

for (i=0; i < CSL_AIF_MAX_NUM_CONTROL_TRANSMISSION_RULES; i++ )
{
   aPeCfg.aCtrlDataTCount[i] = aCtrlTCount[0][i];
   aPeCfg.aCtrlDataCompare[i] = aCtrlCmp[0][i];
   aPeCfg.aCtrlDataSrcSel[i] = aCtrlSrcSel[0][i];
   aPeCfg.aAggrCtrlCtrlDataLut[i] = aCtrlAggrCtrlLut[0][i];
}

//Populate ID LUT for Packet-switched slots
aPeCfg.psDataMask = CSL_AIF_PE_INVALID_PS_DATA_MASK;
aPeCfg.psDataCompare = CSL_AIF_PE_INVALID_COMPARE;
aPeCfg.psDataSrcSel = CSL_AIF_OUT_PS_FIFO_0;
aPeCfg.aggrCtrlPsData = CSL_AIF_PE_AGGR_CTRL_NOP;

//Do the Hw Setup
CSL_aifHwSetup(hAifLink, &aConfigLink);
//Put it in loop back mode
hAifLink[0]->regs->AIF_SERDES0_TST_CFG = 0x40;
ctrlArg = TRUE;
// Enable Transmission
CSL_aifHwControl(hAifLink, CSL_AIF_CMD_ENABLE_DISABLE_RX_LINK,(void*)&ctrlArg);
//Enable Receive
CSL_aifHwControl(hAifLink, CSL_AIF_CMD_ENABLE_DISABLE_TX_LINK,(void*)&ctrlArg);
```

A.5 Troubleshooting

Here are some debugging tips:

- Look at the Rx MAC and Tx MAC registers to see if they are in the proper states. The AIF transmits the data when the Tx MAC is in ST2 and receives the data when Rx MAC is in ST3.
- Check the AI_Ref_Clk and the multiply factor for the SERDES PLL.
- Check the FSYNCCLK/ALTFSYNCCLK and the no of samples per chip for the frame sync setup.
- Check the RX Mac status registers to check for RM_Frame_Bndry_Range error.
- There are some exception event handler registers which are useful in debugging.
- When combining, decombining or redirecting, the C/D and TM frame alignment is very important. This information is available in the C/D and TM status registers.
- For OBSAI packet-switched transfers, the inbound and outbound packet-switched RAM debug memories are useful for viewing the inbound and outbound packet-switched data.

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		**Applications**	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
RFID	www.ti-rfid.com	Telephony	www.ti.com/telephony
Low Power Wireless	www.ti.com/lpw	Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265