

# **Antenna Interface 2 (AIF2) Training**

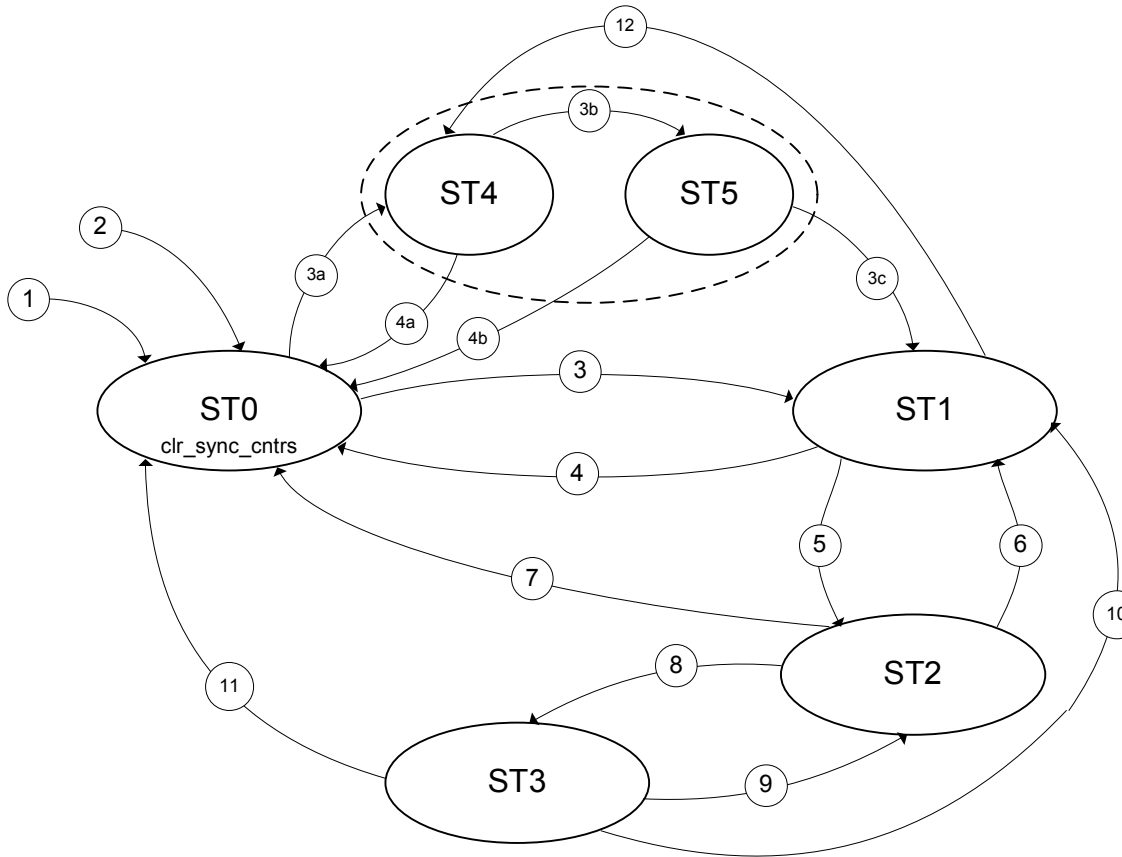
## **Part-3-- advanced topics and usage Examples**

**Brighton Feng**  
**2012-6-18**

# Agenda

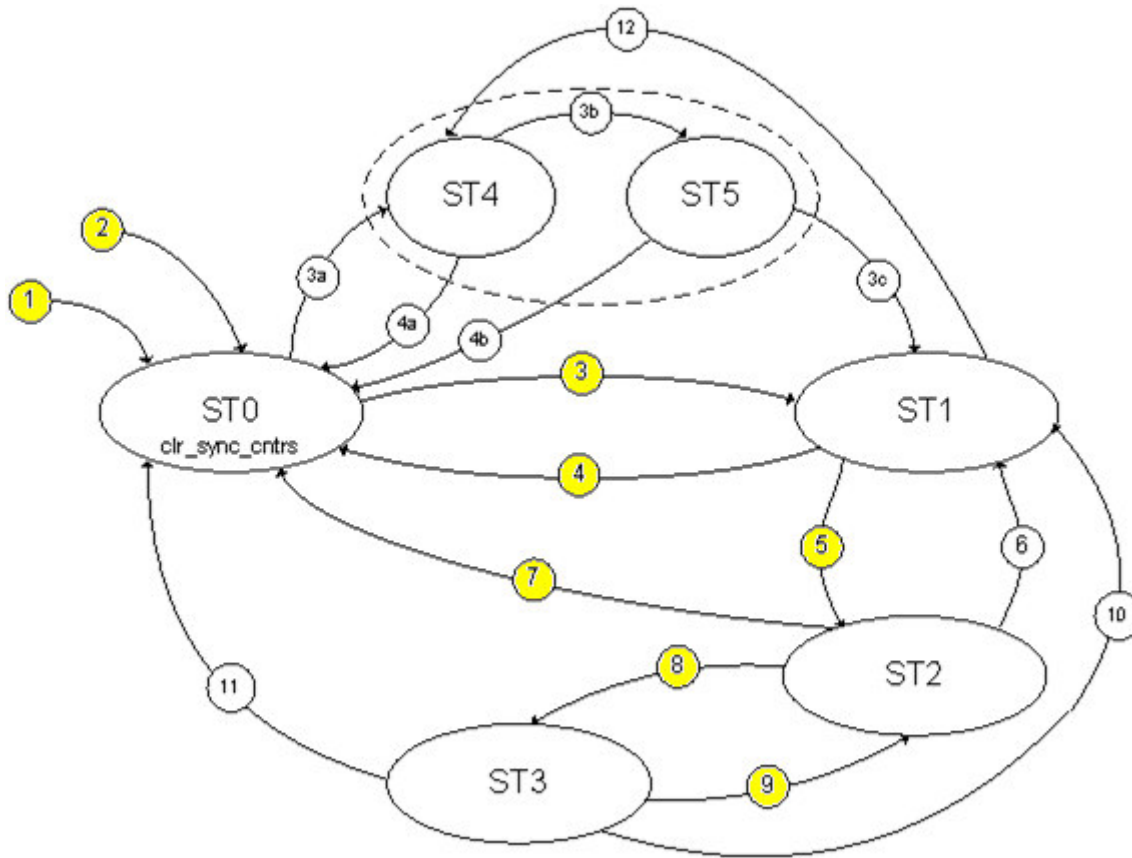
- **Advanced topics**
  - RM State machine
  - TDD configuration
  - Dynamic reconfiguration
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

# OBSAI Rx Mac Receive State Machine



Transition	OBSAI RP3 Description
1	rst_n
2	!rx_en   sd_rm_los_dtct
3a	scr_en & sync_t consecutive valid blocks of bytes received
3b	scr_en & Scrambling seed captured and verified
3c	scr_en & Acknowledge training pattern received
3	!scr_en & sync_t consecutive valid blocks of bytes received
4a	unsync_t consecutive invalid blocks of bytes received
4b	unsync_t consecutive invalid blocks of bytes received
4	unsync_t consecutive invalid blocks of bytes received
5	One K28.7 Idle byte received
6	frame_unsync_t consecutive invalid message groups received
7	unsync_t consecutive invalid blocks of bytes received
8	frame_sync_t consecutive valid message groups received
9	n/a
10	frame_unsync_t consecutive invalid message groups received (Idle order matters)
11	unsync_t consecutive invalid blocks of bytes received OR (lcv_unsync_en & num_los_det)
12	IDEL_REQ pattern detected

# CPRI Rx Mac Receive State Machine



Transition	CPRI Description
1	rst_n
2	<i>!rx_en</i>   LOS (8b10b errors)
3	K28.5 byte received
4	K28.5 byte <i>!</i> received at next Hyperframe boundary
5	K28.5 byte received <b><i>sync_t</i></b> consecutive times at Hyperframe boundary
7	K28.5 byte <i>!</i> received <b><i>unsync_t</i></b> consecutive times at Hyperframe boundary
8	K28.5 byte received <b><i>frame_sync_t</i></b> consecutive times at Hyperframe boundary
9	K28.5 byte <i>!</i> received <b><i>frame_unsync_t</i></b> consecutive times at Hyperframe boundary

# Agenda

- **Advanced topics**
  - RM State machine
  - TDD configuration
  - Dynamic reconfiguration
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

# TDD configuration

- **TDD mode can be supported by packet DMA, it is implemented in following ways:**
  - TX: only sends packet in active TDD time slots
  - RX: only receive packets in active TDD time slots

# TDD TX configuration

- Enable TDD\_AXC mode. PE will insert symbol only when it receives the packet from packet DMA.
- Do not push packets into the packet DMA TX queue during inactive TDD time slots
- A packet should only be pushed into TX queue just one packet ahead, for example, if packet of LTE symbol N need be sent in active TDD time slot, then packet N must be pushed into the TX queue in the time period of symbol (N-1).

# TDD RX configuration

- PD implements programmable bit map for each channel, each bit corresponds to one packet in a radio Frame, “1” means corresponding packet will be received, “0” means that packet will be dropped.
- Please note, in LTE protocol level, the TDD is controlled in the unit of sub frame, but the AIF2 PD TDD bit map is defined in the unit of packets, and one LTE sub frame includes multiple packets (symbols), so the LTE TDD “bit map” in sub frame should be expanded to PD TDD “bit map” in packets, that is, we need to program multiple bits to select each LTE symbol in a sub frame.

# Agenda

- **Advanced topics**
  - RM State machine
  - TDD configuration
  - **Dynamic reconfiguration**
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

# Dynamic Configuration

Types of changes are split into two basic categories:

- On-the-fly: Ping-pong configuration mechanism allows change to occur from one frame to the next without any “off” or error period of time.
  - Change TDD bit map
  - Change Symbol size (PD/PE Framing counter)
- Normal Changes: torn down and then later rebuilt to accomplish a change. These changes require a “system down” period (normally, it is about 3 radio frames).
  - AxC reconfiguration: change AxC offset
  - Link reconfiguration: change DBM (antenna bandwidth), change timing

# AIF2 Software Reset

- AIF2 software reset can be used when whole AIF need be reconfigured.
  - A single MMR contains a bit which is used as the software-controlled hardware reset of the AIF2.
  - AIF2 CSL supports API called `CSL_aif2Reset(hAif2)`, which activates a software reset process.
  - The entire AIF2 hardware can be reset except for following circuitry is NOT reset during software reset:
    - Config VBUSP Interface
    - Internal SCR circuits

# Agenda

- **Advanced topics**
  - RM State machine
  - TDD configuration
  - Dynamic reconfiguration
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

# CPRI Protocol Specific

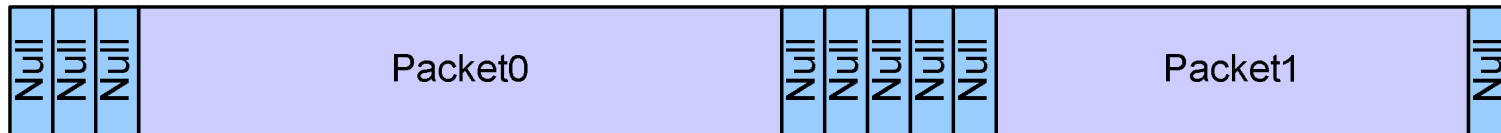
## CPRI Control Data

- Slow C&M (HDLC) ← Not Supported by AIF2
- Fast C&M (Fast Ethernet) ← Supported and extended by AIF2
- CPRI supports 4b/5b encoding, decoding and Ethernet packet inserting and parsing
- Adding or stripping SOP (start of packet) data from/into the frame data
- Using four channels to support CPRI control stream

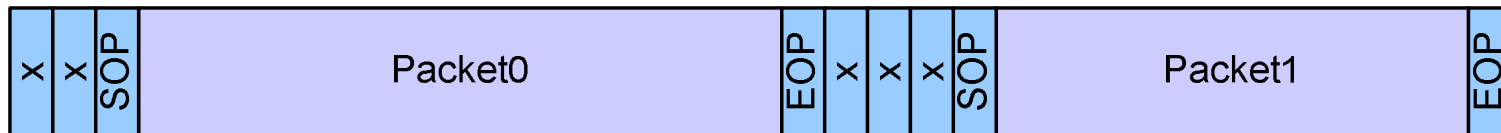
## Packet Parsing

- Two mechanisms for packet parsing:
  - Programmable Null delimiter (eg. K27.7 or K29.7)
  - 4B/5B encoding/decoding
- Each of four possible (per link) control streams are configured as to which of the two options is used for that stream .

### Null Delimited Packets



### 4B/5B Delimited Packets



# OBSAI Protocol Specific: Time Stamp

## AxC Data Time Stamp

The 6 bits of time stamp start at 6'b000000 at the radio frame boundary and increment by +1 every OBSAI message. The time stamp value ranges from 0-to-63.

## Ethernet Time Stamp

SOP: 6'b100000

MOP: 6'b000000

EOP: 6'b1XXXXX (XXXXX: indicates the number of bytes from the start of RP3 payload containing MAC frame data (counting started from the byte after the header))

## Generic Packet Time Stamp

SOP: 6'b10XXXX

MOP: 6'b00XXXX

EOP: 6'b11XXXX (XXXX: is an extension of OBSAI address and is the same for all elements within the packet)

# Generic data transfer over AIF2

	OBSAI	CPRI
Over control message slots or control words	<p>Bandwidth utilization is about 1/21.            For 8x link, bandwidth is about <math>6.144 \times 0.8 \times (16/19) \times (399/400) / 21 =</math>  <b>196Mbps</b></p>	<p>Can use 4B/5B or NULL delimiter. Bandwidth utilization is about <math>1/16 \times (64-P)/64</math>, P is the start pointer of generic data in the control words array.            For 8x link, assume NULL delimiter, P=20, bandwidth is about <math>4.9152 \times 0.8 / 16 \times 44/64 =</math> <b>168Mbps</b></p>
Over AxC slots	<p>Bandwidth utilization is about 20/21            For 8x link, bandwidth is about <math>6.144 \times 0.8 \times (16/19) \times (399/400) \times (20/21) =</math>  <b>3.9Gbps</b></p>	<p>4B/5B delimiter must be used; Bandwidth utilization is about <math>(4/5) \times (15/16)</math>.            For 8x link, bandwidth is about <math>4.9152 \times 0.8 \times 15/16 \times (4/5) =</math> <b>2.9Gbps</b></p>

- OBSAI mode supports generic data transfer better.

# Agenda

- **Advanced topics**
  - RM State machine
  - TDD configuration
  - Dynamic reconfiguration
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

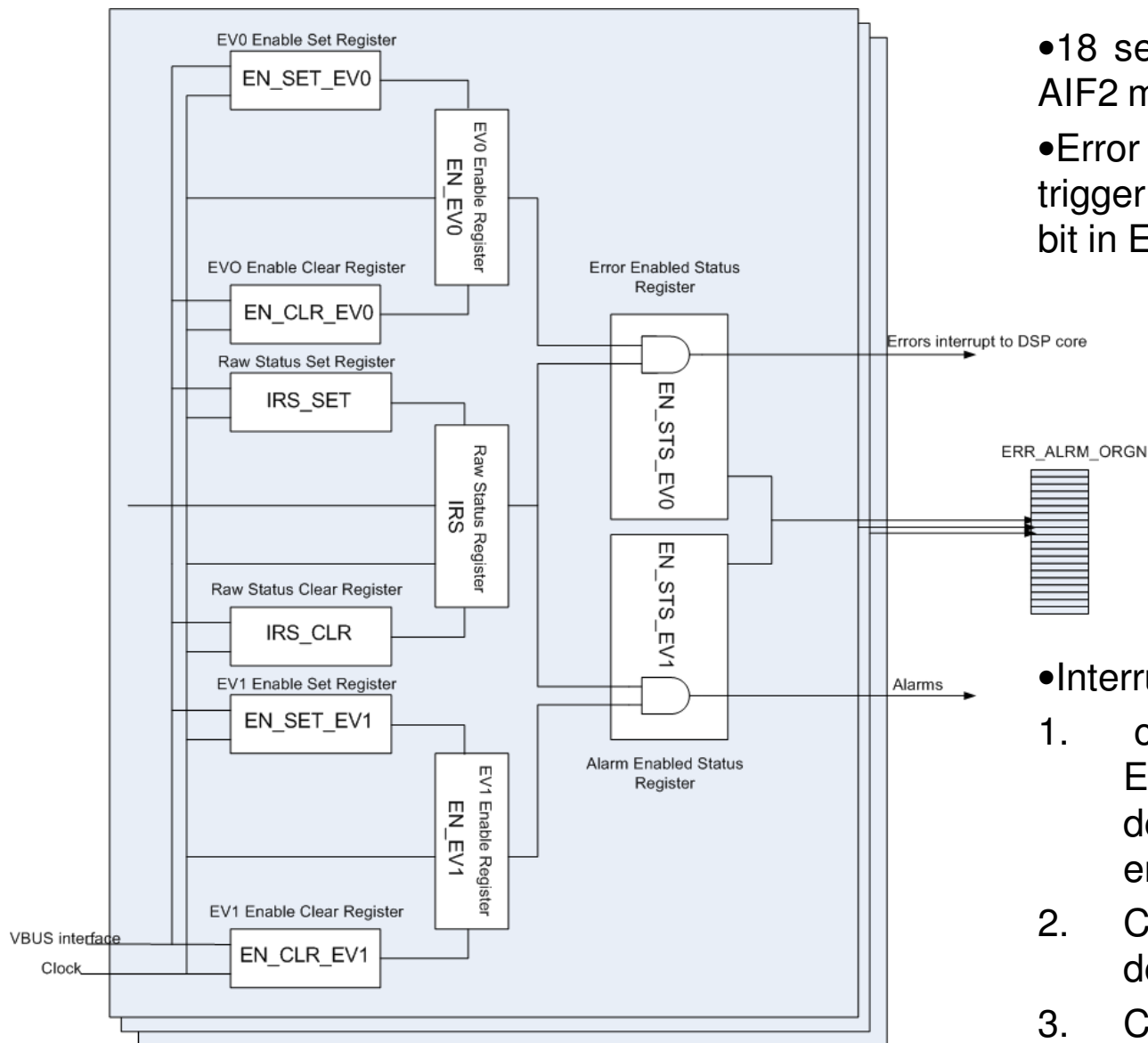
# Debug tips

- Some common debug methods, such as breakpoint, step through, or printf() during AIF2 running, may affect the operation of AIF2, because AIF2 is very timing sensitive, once these debug operations happen, AIF2 timing may be broken, so, after these operations, AIF2 may not function properly.
- Some AIF2 status registers are volatile after normal data transfer, when you manually check them, they may have changed.
- So, the best way is to dump AIF2 status registers to a data buffer or structure by software right after the expected data transfer completes. And then, user can check the data structure manually, or software can print these statuses later.

# **AIF2 statuses helpful for debug**

- **RM/TM state**
- **AD EOP counter (24-bit), this counter will wrap when it reaches its maximum value.**
  - In packet DMA mode, it counts the packet number.
  - In DIO mode, it counts the ingress data burst only; egress DIO data is not counted.
- **AT frame, symbol/slot, clock count.**
- **Captured RM Pi offset**
- **All other error/status**

# EE (Error and Exception) Module



- 18 sets of registers, each for one AIF2 module.
- Error happens in any module will trigger interrupt, and corresponding bit in ERR\_ALARM\_ORGN is set.

## Interrupt Service Routine should:

1. check the ERR\_ALARM\_ORGN to determine which module has error
2. Check the EN\_STS to determine what the error is.
3. Clear the IRS

# Common errors and possible reasons

- **RM Line Code Violations, or RM not in SYNC state.**
  - Normally, this is because hardware signal integrity is not good, or the other side of the AIF link does not run properly.
- **PI out of window**
  - Normally, this is because the AIF2 event timing is not configured properly.
- **data shift**
  - Normally, this is because the AIF2 AxC offset is not configured properly.
- **PE DB did not have data for a channel**
  - Normally, this is because the packet DMA does not transfer data in time.
- **PE Symbol index in Navigator protocol specific header did not match for one or more symbol.**
  - Normally, this is because the packet is not pushed into the TX queue in correct time, or the symbol index in the protocol specific header is not set correctly.
- **Packet DMA descriptor starvation.**
  - Normally, this is because the descriptors are not returned to the FDQ in time, or RX data is generated faster than processing.

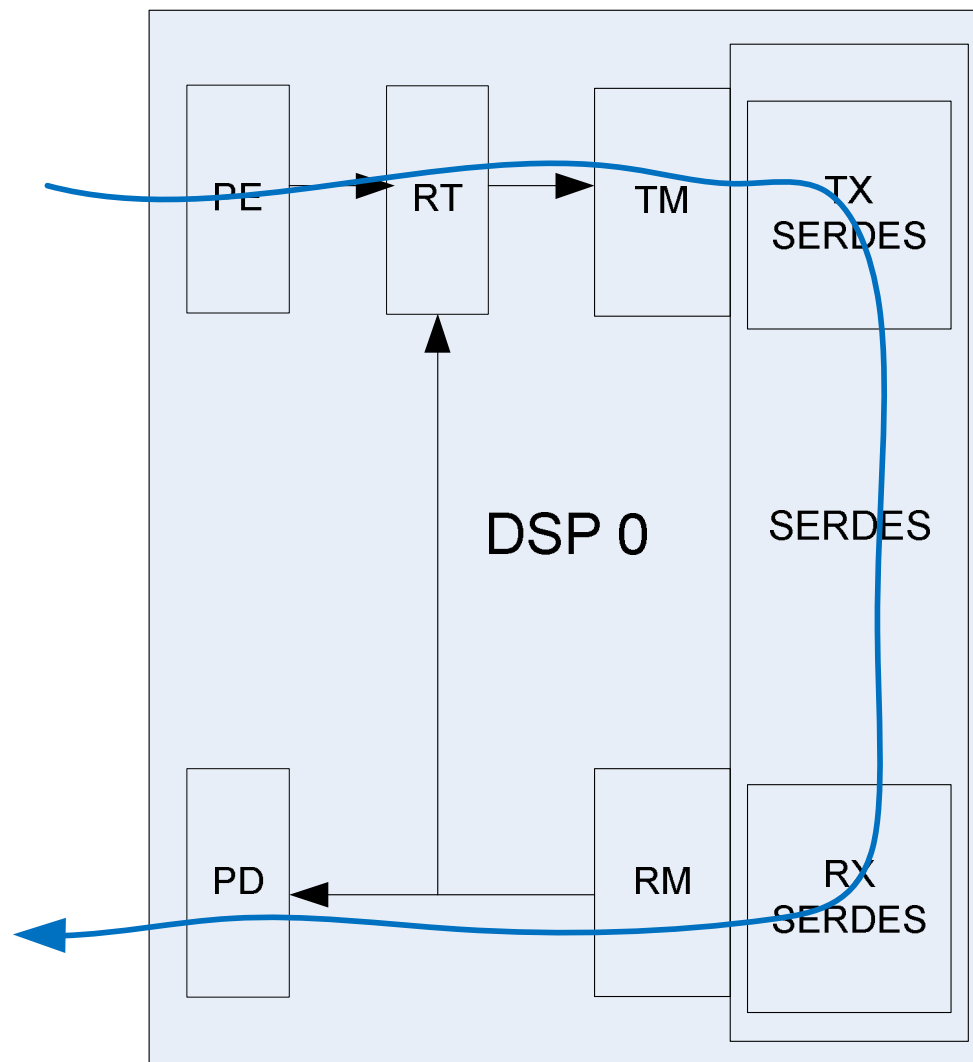
# Agenda

- **Advanced topics**
  - RM State machine
  - TDD configuration
  - Dynamic reconfiguration
  - Control/Generic data transfer
  - debugging tips and Frequently encountered issues
- **AlF2 usage examples**

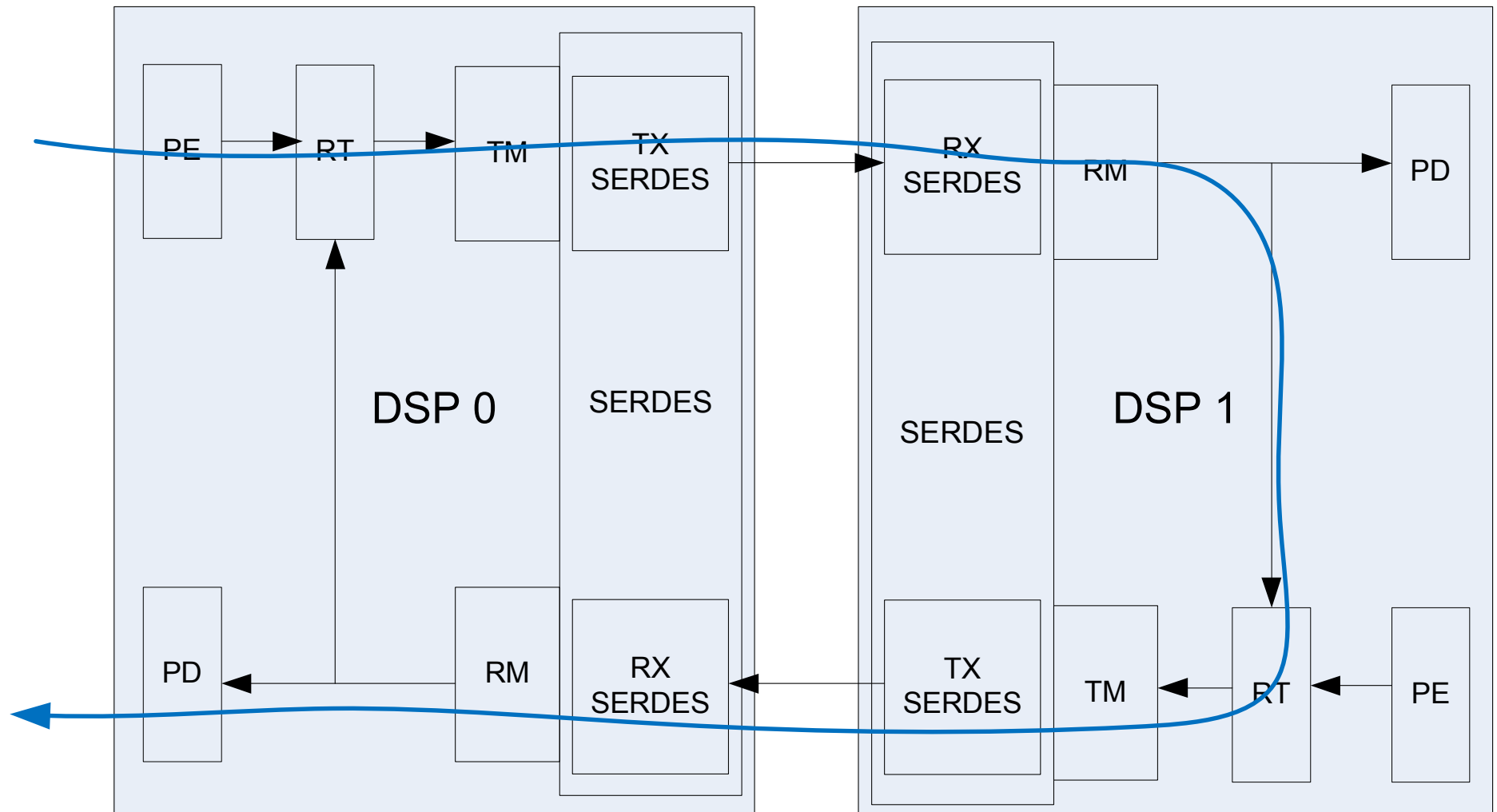
# Usage examples

- **Examples are provided to show following usage cases:**
  - Protocol: OBSAI, CPRI
  - Radio Standard: LTE (FDD/TDD, normal/extended symbol), WCDMA, TD-SCDMA
  - Link rate: 2x, 4x, 8x
  - Change data buffer between LL2, SL2, DDR
  - Data Type
    - Antenna data only (on AxC slots)
    - Generic data only (on AxC slots)
    - Antenna data (on AxC slots) and generic data (on control slots)
  - Data Path
    - Internal loopback
    - External redirection (between 2 DSPs)

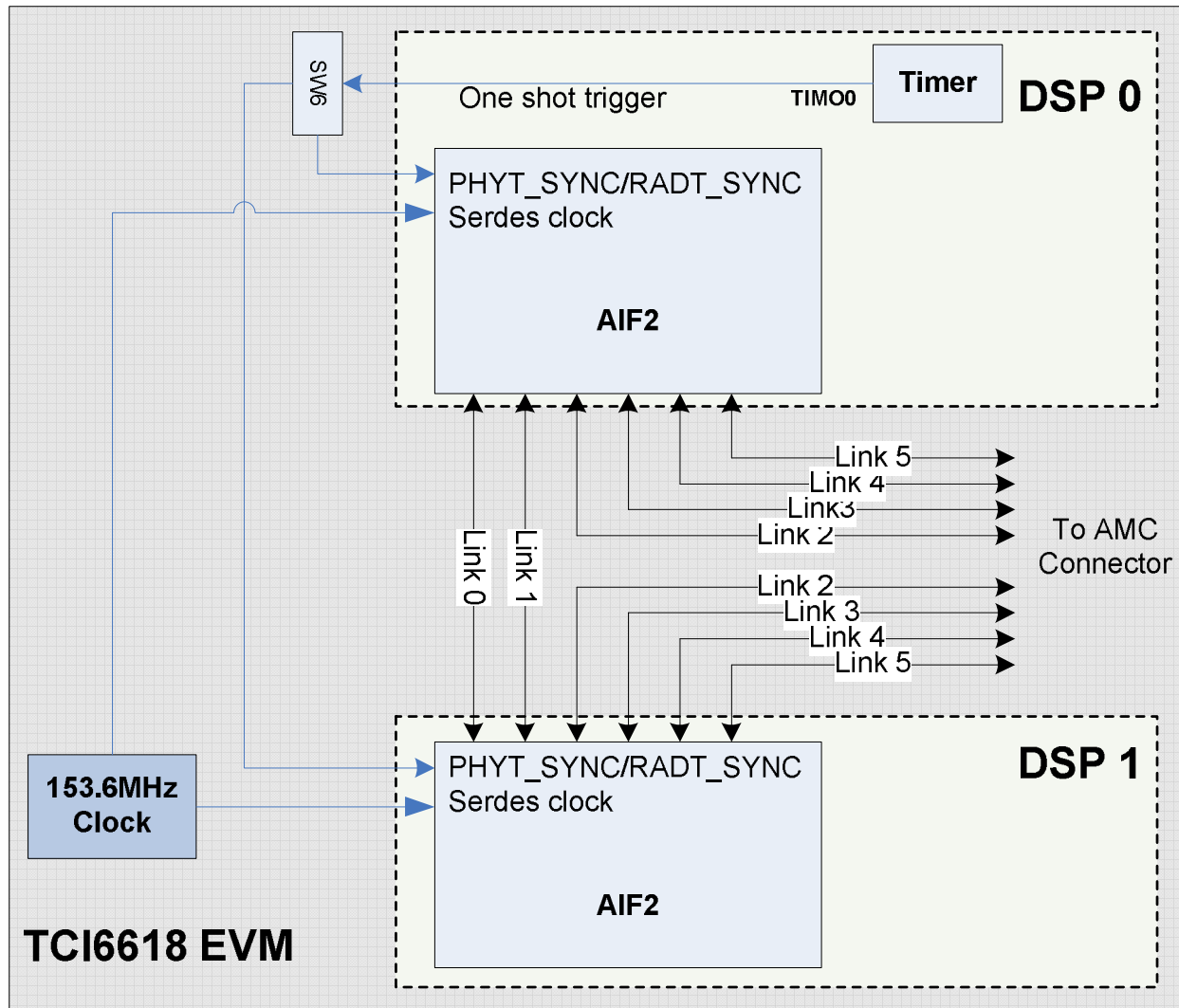
# Internal loopback test



# External redirection test

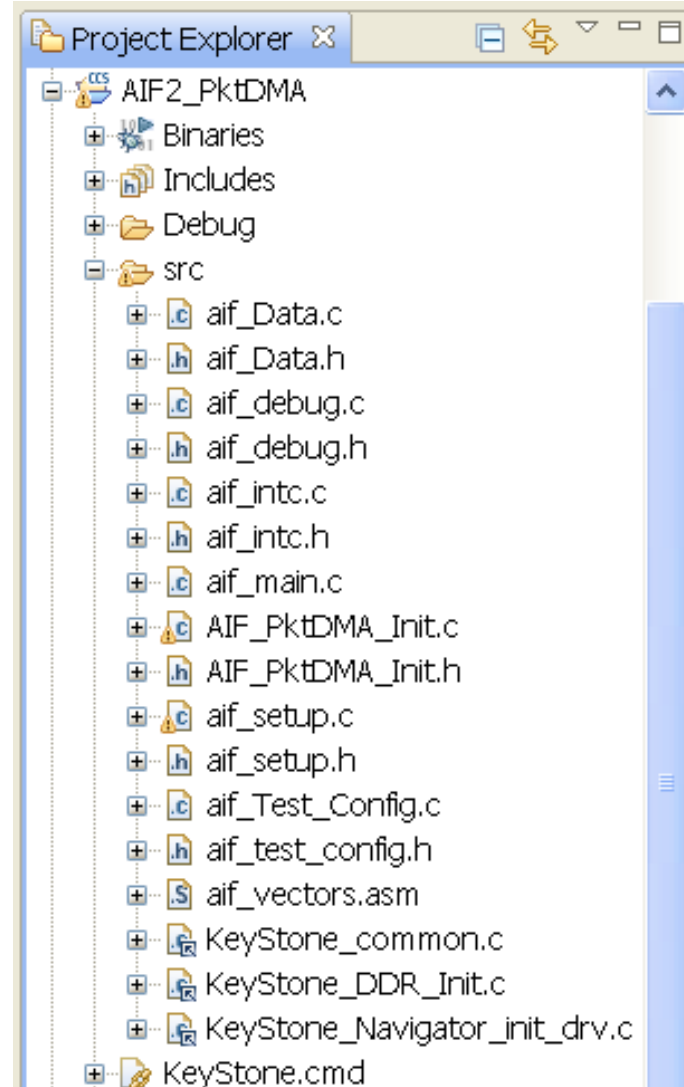
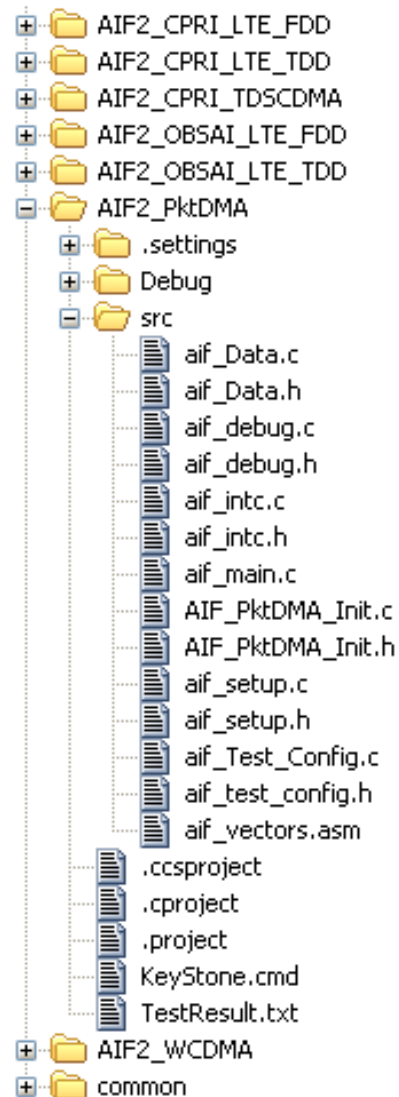


# Dual TCI6618 EVM for redirection test



- DSP0 timer is configured to generate one shot pulse, which triggers both AIF2 on the DSP0 and DSP1 to run simultaneously. AIF2 timers on both DSP are configured to run freely, external periodical frame synchronization signal is not required for this test.

# Test projects



# Source files of the example codes

Files	Descriptions
aif_main	main() function and the top level control codes
aif_Test_Config	This file includes basic configuration structure for test. User can modify the basic parameters to change the test mode to verify most functionalities of AIF. The codes in this file calculate many other parameters for AIF based on the basic configuration.
aif_setup	Setup AIF registers according the configuration parameters from aif_Test_Config module.
aif_Data	This file includes code for sending data and processing RX data TX/RX packets statistics information is collected and summarized
aif_PktDMA_Init	This file setup Packet DMA and QMSS for AIF2.
aif_intc	This file setup interrupt controller, and includes ISR for Frame/Slot/Symbol, data transfer are triggered by them. ISR of QMSS accumulation triggers RX packet processing. ISR for error log.
aif_debug	This file include functions to log and print status and error for debug.

# To run the examples on TI's EVM

- The examples can be run on Nyquist and Appleton EVMs:
  - To run the redirection test between two DSPs, run DSP1 firstly, then run DSP0.
- Test configuration can be changed in “aif\_Test\_Config.c” as below. To rebuild the project with your new configurations on your PC, you may need to change the CSL include directory.

```
/*Configuration for link 0*/
```

```
{
```

```
    1,          /*Link Enable: 1=enable, 0=disable*/
```

```
    CSL_AIF2_LINK_RATE_8x,
```

```
    TEST_PATH_INTERNAL_LOOPBACK,          /*test data path*/
```

```
    AIF2_AXC_DATA_ONLY,          /*testDataType*/
```

```
    AIF_MONO_PACKET_SIZE-16,          /*genericPacketSize*/
```

```
    0,          /*numberAxC: 0 means maximum number*/
```

```
    LTE_20MHZ          /*lteBandwidth*/
```

```
},
```

# Demo show

CCS Debug - AIF2\_CPRI\_LTE\_FDD\_Test/src/aif\_main.c - Code Composer Studio

File Edit View Search Project Tools Run Scripts Window Help

Debug

AIF2\_CPRI\_LTE\_FDD\_Test [Code Composer Studio - Dev]

- Blackhawk XDS560v2-USB Mezzanine Emulator\_0/C66x
- Blackhawk XDS560v2-USB Mezzanine Emulator\_0/C66x
- Blackhawk XDS560v2-USB Mezzanine Emulator\_0/C66x
- Blackhawk XDS560v2-USB Mezzanine Emulator\_0/C66x

aif\_main.c aif\_Data.c aif\_setup.c aif\_debug.c aif\_intc.c aif\_Test\_Config.c

```
66
67 void main()
68 {
69     int i;
70
71     #if 1 /*for debug only*/
72         //disable AIF through PSC
73         Keystone_disable_PSC_module(CSL_PSC_PD_AI, CSL_PSC_LPSC_AI);
74         Keystone_disable_PSC_Power_Domain(CSL_PSC_PD_AI);
75     #endif
76 }
```

Console Memory Map

AIF2\_CPRI\_LTE\_FDD\_Test:CIO

```
[C66xx_1] Initialize DSP main clock = 122.88MHzx236/29 = 999MHz
[C66xx_1] Initialize DDR speed = 66.667x20/1= 1333.3
[C66xx_1] =====AIF CPRI mode test for 5000 ms (500 frames, LTE FDD normal
[C66xx_1] link 0 runs at 4x rate, external line loopback test, antenna data in AxC slot or
[C66xx_1] link 1 runs at 4x rate, internal loopback test, antenna data in AxC slot or
[C66xx_1] link 2 is disabled
[C66xx_1] link 3 is disabled
[C66xx_1] link 4 is disabled
[C66xx_1] link 5 is disabled
```

Registers Variables

Name	Type	Value
------	------	-------

Expressions Memory Browser

Expression	Value
errLog	0x82A7CA70 (Hex)
uiSlotCount	33
uiFrameCount	3
uiSymbolCount	88
uiTestNumFrame	50000
aif2Regs->EE_LK[0].EE	0x00000000 (Hex)
aif2Regs->EE_DB_EN	0x00000000 (Hex)
aif2Regs->EE_AD_EN	0x00000000 (Hex)
aif2Regs->EE_CD_EN	0x00000000 (Hex)
aif2Regs->EE_PD_COM	0
aif2Regs->EE_PE_COM	0
aif2Regs->EE_SD_EN	0x00000000 (Hex)
pktDmaRxChCfgRegs->	Error: identifier not f
err_sts_log_cnt	5
aifLinkCfg	0x008BBC30
aif2Regs->DB_IDB_PT	0x01F10200 (Hex)
aif2Regs->DB_EDB_PT	0x01F11200 (Hex)
aif2Regs->PE_AXC_OF	0x01F6C800 (Hex)
uiEmptyFlag	Error: identifier not f
uiNumAifChannel	0
+ Add new expression	

Licensed BE

# To run the examples on different board

- on other board, if timer is not used to trigger the AIF2, other method must be used. For example, for simple test, the AIF2 can be triggered by manually writing specific registers. This can be enabled by the macro defined in the “aif\_Test\_Config.h”:

```
//#define AIF2_TRIGGER_MANUALLY
```

- on other boards, the reference clock for AIF2 may be different, this can be configured with following code in “aif\_setup.c”

```
#ifdef AIF2_LINK_PROTOCOL_OBSAI
```

```
    pllMpy= CSL_AIF2_PLL_MUL_FACTOR_20X; /*153.6*20=3072*/
```

```
#else
```

```
    pllMpy= CSL_AIF2_PLL_MUL_FACTOR_16X; /*153.6*16=2457.6*/
```

```
#endif
```

- DSP core PLL and DDR configuration may need be changed in “aif\_main.c”, or GEL or other methods should be used to initialize them.

```
//DSP core speed: 122.88*236/29= 999.9889655MHz
```

```
KeyStone_main_PLL_init(122.88, 236, 29);
```

```
//DDR init 66.667*20/1= 1333
```

```
KeyStone_DDR_init (66.667, 20, 1);
```

# Q&A

# Multiple choice questions from Part-3

1. **The unit of CPRI RM synchronization threshold is:**
  - A. Byte
  - B. Basic frame
  - C. Hyper frame
  - D. Radio frame
2. **Which parameter can be reconfigured on-the-fly:**
  - A. TDD bit map
  - B. AxC offset
  - C. DBM rules
  - D. Delta
3. **Choose all true statements for generic data transfer over AIF2:**
  - A. CPRI mode supports two generic packet delimitation mechanisms: NULL, 4B/5B
  - B. OBSAI mode delimitates generic packets with time stamp in message header
  - C. OBSAI mode supports generic data transfer better.
  - D. Generic data can not be transferred over AxC slot.

# Multiple choice questions from Part-3

## 4. Choose true statement about AIF2 error and debugging:

- A. CIO functions, such as printf(), can be freely used during AIF2 running.
- B. EOP counter increases with egress DIO transfer.
- C. In interrupt service routine, error flag in IRS register should be cleared to enable additional interrupt.

## Choose possible reason for following errors:

5. RM Line Code Violations, or RM not in SYNC state. **D**

6. data shift **B**

7. PE DB did not have data for a channel **C**

8. Packet DMA descriptor starvation. **A**

- A. descriptors are not returned to the FDQ in time, or RX data is generated faster than processing.
- B. AxC offset is not configured properly.
- C. the packet DMA does not transfer data in time.
- D. hardware signal integrity is not good, or the other side of the AIF link does not run properly.