Customer Training Workshop

Traveo™ II Clock Extension Peripheral Interface (CXPI)

Q4 2020

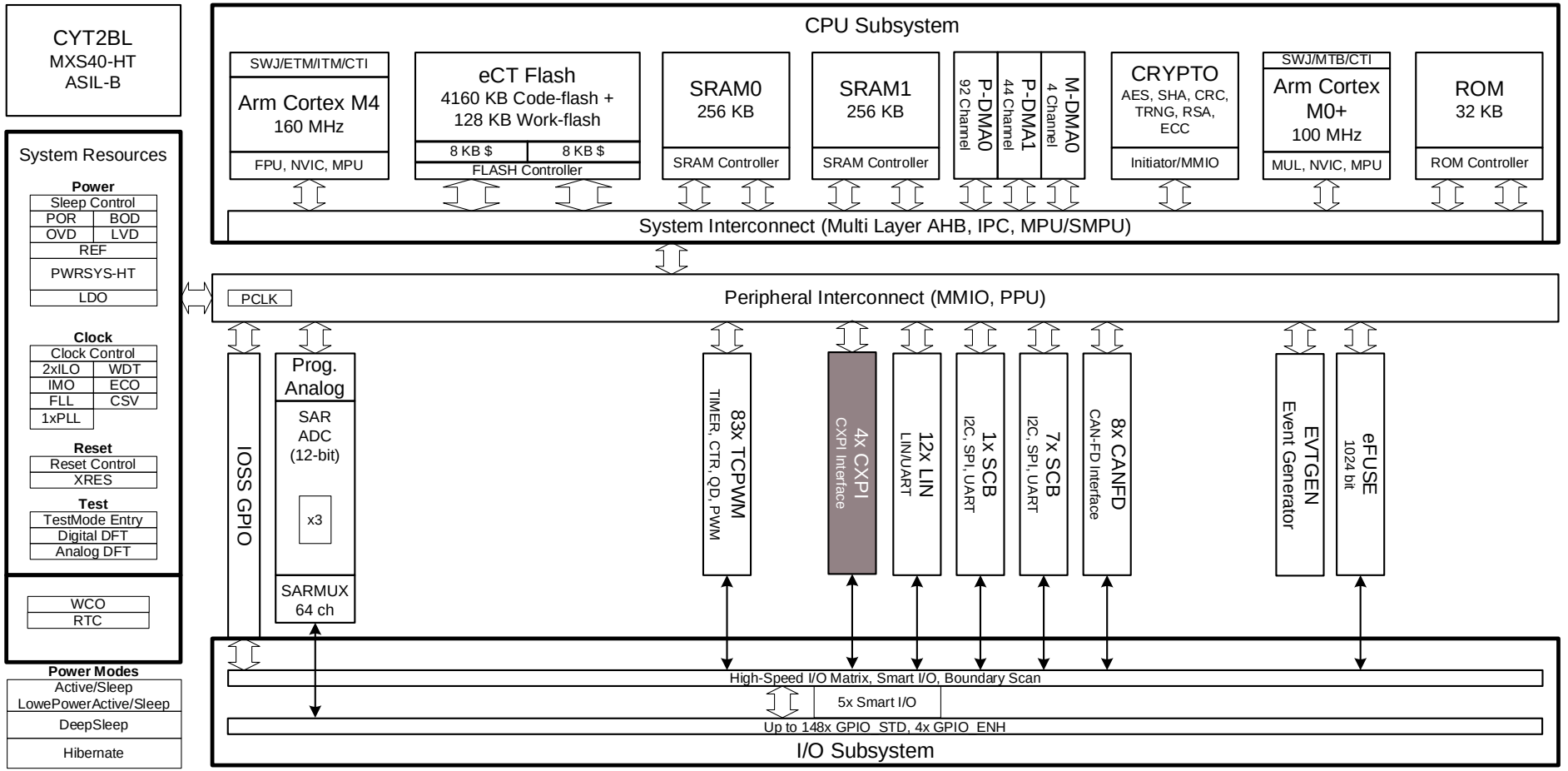
Target Products

› Target product list for this training material:

Family Category	Series	Code Flash Memory Size
Traveo™ II Automotive Body Controller Entry	CYT2B9	Up to 2112KB
Traveo™ II Automotive Body Controller Entry	CYT2BL	Up to 4160KB
Traveo™ II Automotive Cluster	CYT3DL	Up to 4160KB
Traveo™ II Automotive Cluster	CYT4DN	Up to 6336KB

Introduction to Traveo II Body Controller Entry

› The CXPI controller is part of Peripheral Blocks

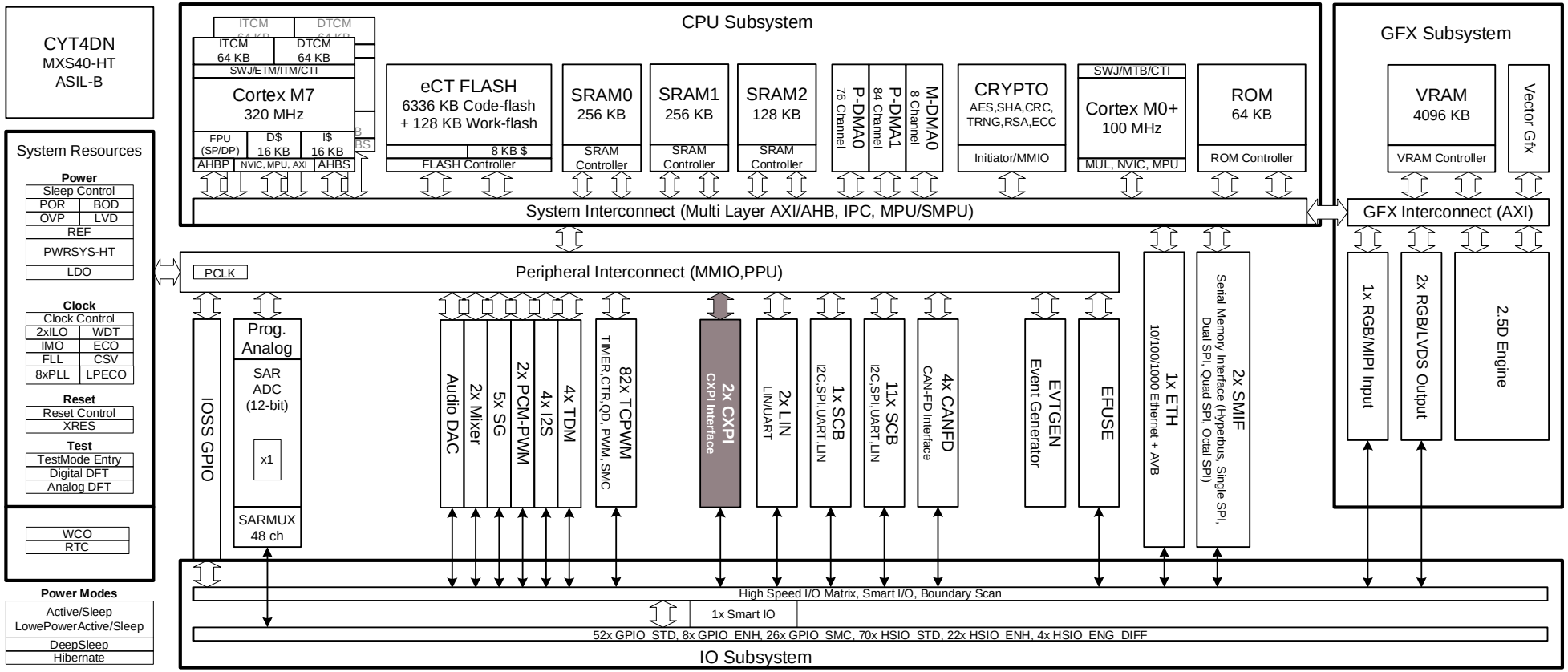


Hint Bar

Review TRM chapter 30 for additional details

Introduction to Traveo II Cluster

› The CXPI controller is part of Peripheral Blocks



Hint Bar

Review TRM chapter 30 for additional details

CXPI Overview

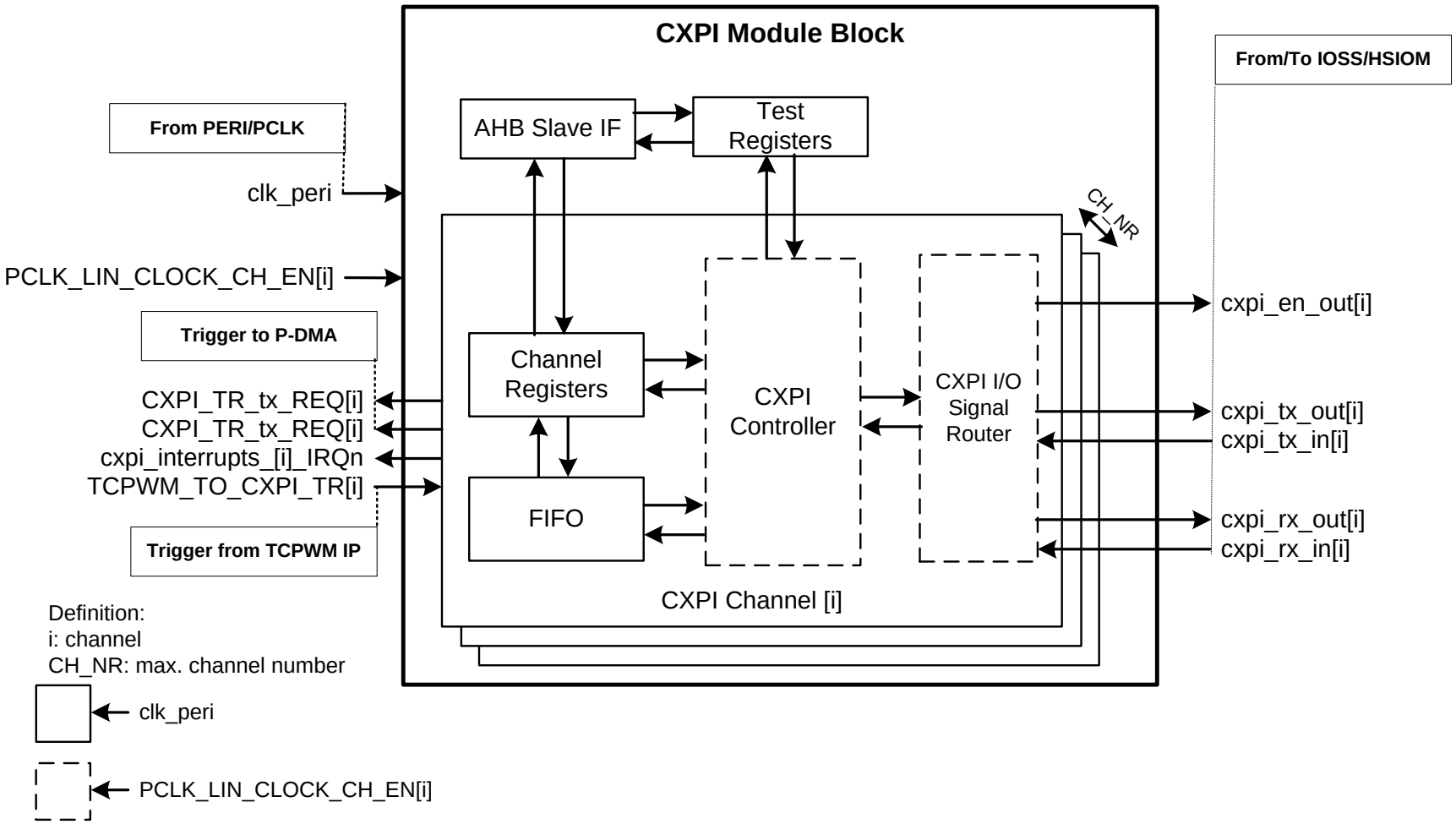
- Clock Extension Peripheral Interface (CXPI) is a single-line communication bus with clock modulation to synchronize all slave nodes with the master clock
- Features
  - CXPI protocol support in hardware according to ISO/WD 20794-4
  - Master and Slave nodes
    - Autonomous request field and response transfer processing
  - Data signal encoding and decoding formats
    - Non-return to zero (NRZ) mode
    - Pulse-width modulation (PWM) mode
  - Wake pulse generation
  - Error detection
  - Timeout detection
  - Test modes including hardware error injection

Hint Bar

Review TRM section 30.1 for additional details

CXPI Block Diagram

› CXPI module block component

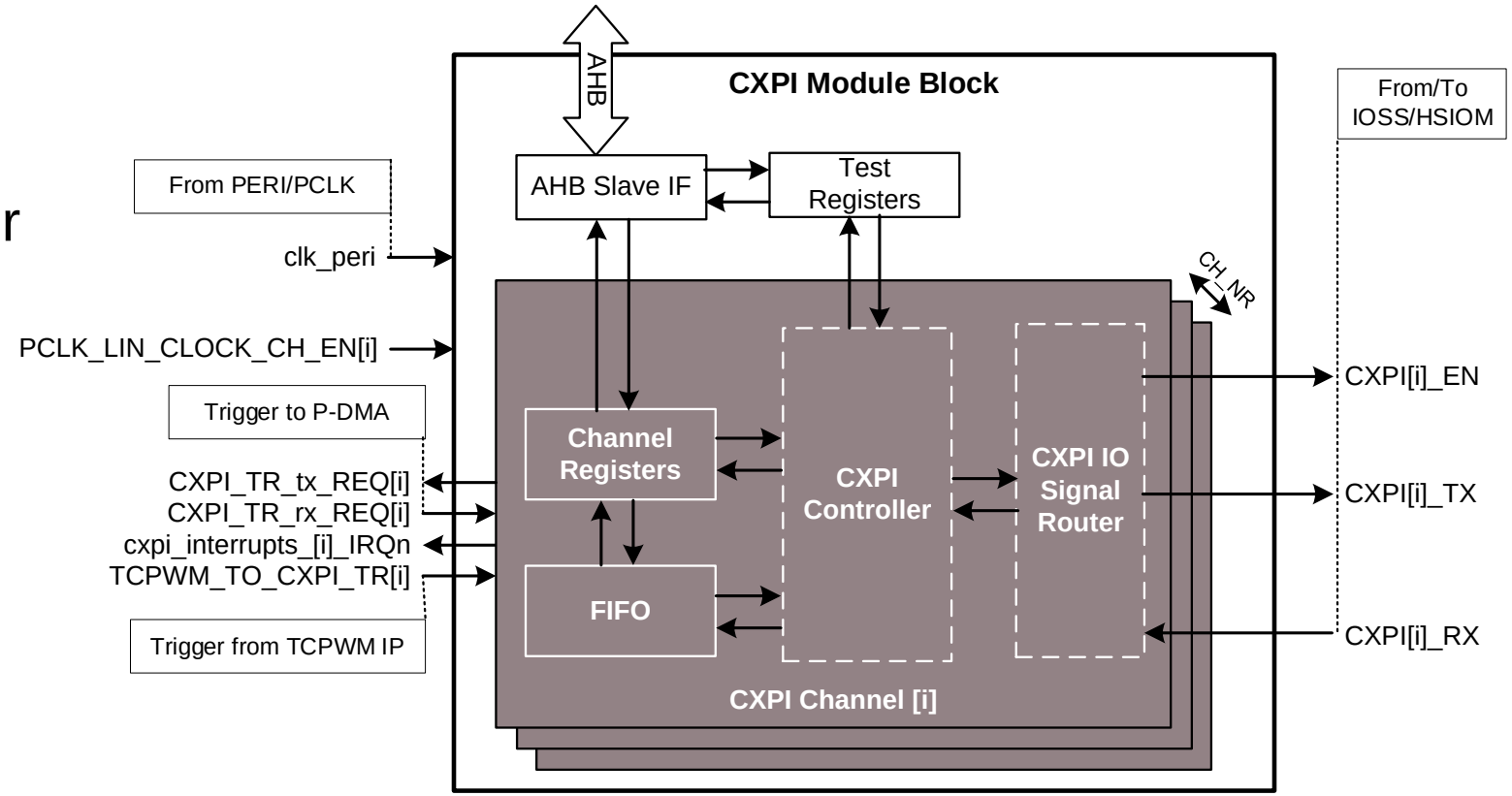


Hint Bar

Review TRM section 30.2 for additional details

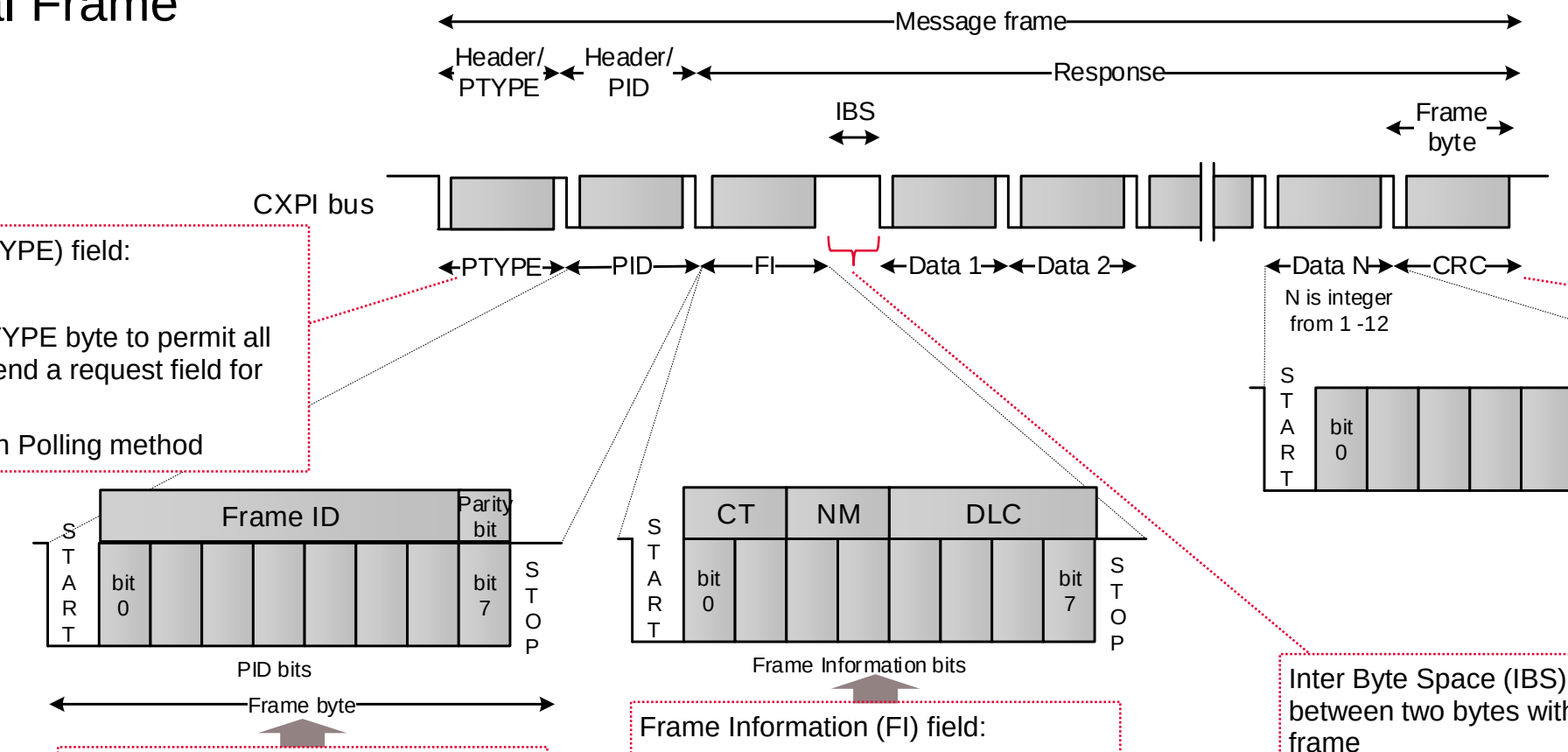
CXPI Module Block Components

- CXPI Channel
  - Message Frame Format
  - Message Transfer Processing Commands
  - Message Frame Transfer
  - Arbitration Loss
  - Bus Signal Modulation
  - FIFO Buffer
  - P-DMA Trigger
  - Timeout Detection
  - Enabling CXPI Channel
  - Power Modes
  - Oversampling
  - Noise filter



Message Frame Format (1/2)

› Normal Frame



Protected Type (PTYPE) field:

- 8-bit of PTYPE
- Master sends PTYPE byte to permit all slave nodes to send a request field for this time slot
- Only applicable in Polling method

CRC field: 8 bits

Inter Byte Space (IBS): The idle time between two bytes within a message frame

Protected Identifier (PID) field:

- 7-bit frame identifier
- 1-bit odd parity over frame identifier

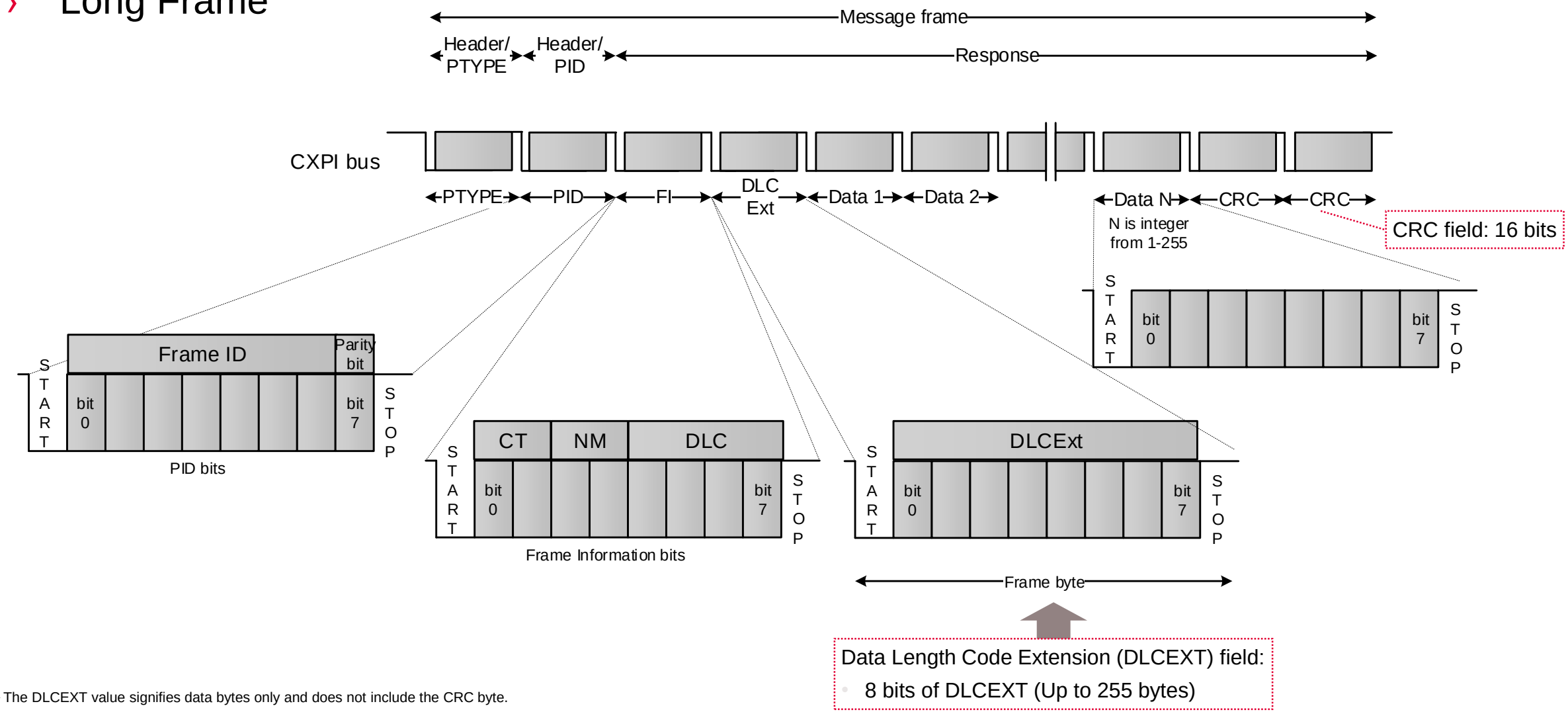
Frame Information (FI) field:

- 2 bits Counter (CT)
- 2 bits Network Management (NM)
- 4 bits Data Length Code (DLC) (Up to 12 bytes)

[1] The DLC value signifies data bytes only and does not include the CRC byte.

Message Frame Format (2/2)

› Long Frame



[1] The DLCEXT value signifies data bytes only and does not include the CRC byte.

Message Transfer Processing Commands

- These commands are set in the CMD register

Command	Description
TX_HEADER	Transmit a header
TX_RESPONSE	Transmit a response
IFS_WAIT	Check if the bus is in idle state before transmitting
RX_HEADER	Receive a header
RX_RESPONSE	Receive a response

- Advantage
  - Reduces CPU load

Hint Bar

Review TRM section 30.5.8 and Register TRM for additional details

Inter Frame Space (IFS)

Message Frame Transfer

- Method 1: Event Trigger Method
  - The master and each slave node can start a frame when the bus is ready for transmission
  - Advantage:
    - High responsivity to the event
- Method 2: Polling Method
  - Message transfers are scheduled in general and thereby every transfer is initiated by the master
  - Advantage:
    - Ensures the periodicity of the communication

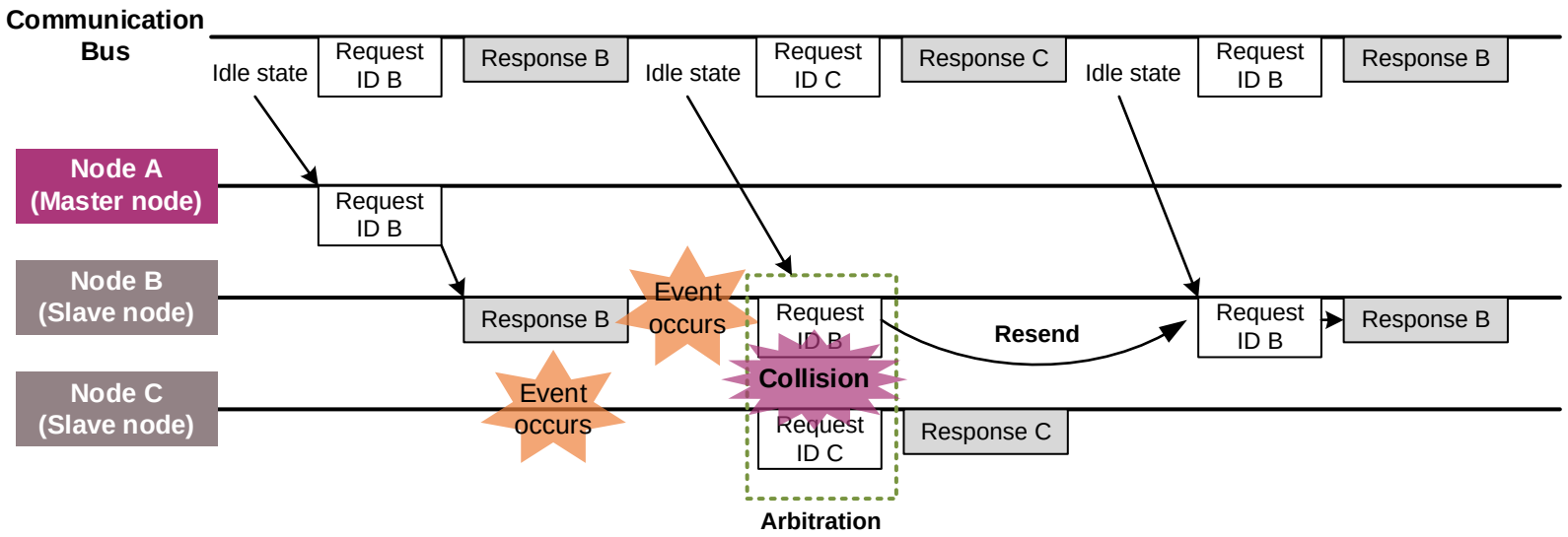
Hint Bar

Review TRM section 30.5.1 for additional details

Either method can be chosen and implemented according to the system requirements

Method 1: Event Trigger Method Example

① Node A transmits Request ID B after detecting that the bus is in idle state
② Node B, corresponding to Request ID B, transmits Response B
③ Event occurs on node B and node C
④ Request ID B and Request ID C are transmitted at the same time when the bus is in idle state
⑤ Collision occurs
⑥ Request ID C (node C) is transmitted as a result of arbitration and Response C is retransmitted
⑦ Node B transmits Request ID B based on the time when the bus moves to idle state



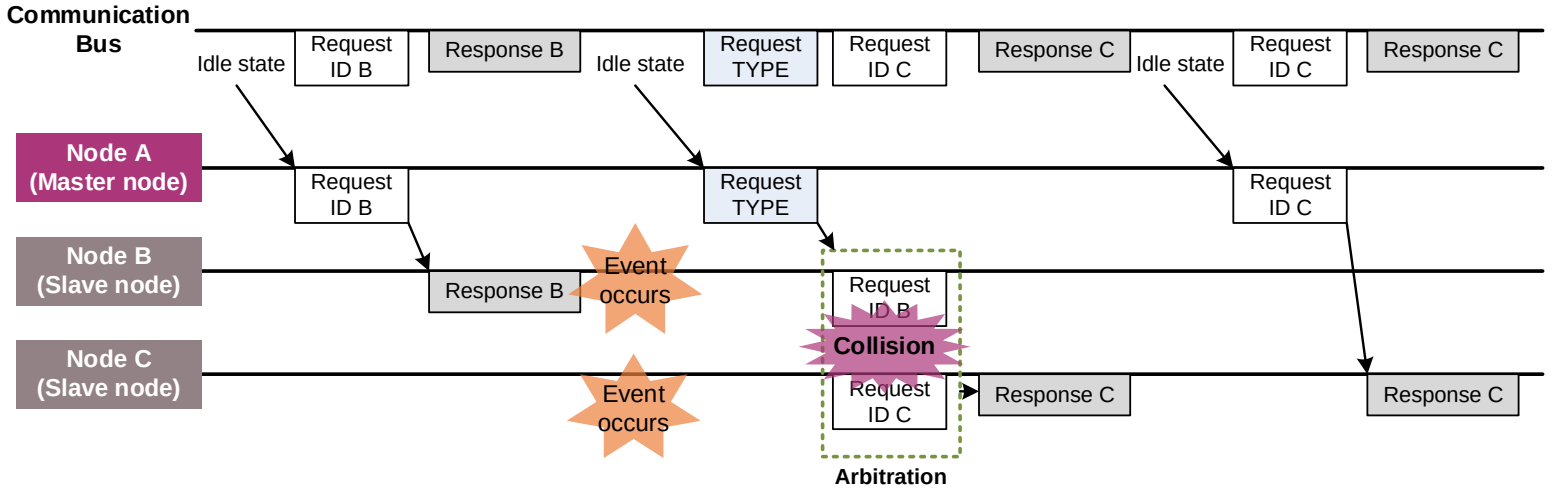
Hint Bar

Review TRM section 30.5.1 for additional details

Arbitration takes place according to PID field priority. The smaller frame ID with the bit of logical value '0' has priority over the bit of logical value '1'

Method 2: Polling Method Example

① Node A transmits Request ID B after detecting that the bus is in idle state
② Node B, corresponding to Request ID B, transmits Response B
③ Event occurs on node B1 and node C
④ Node A sends a PTYPE field to permit all slave nodes to send a request field for this time slot
⑤ Request ID B and Request ID C are transmitted at the same time
⑥ Collision occurs
⑦ Request ID C (node C) is transmitted as a result of arbitration and Response C is retransmitted
⑧ Node C transmits response so that node A transmits frame ID C of the node C



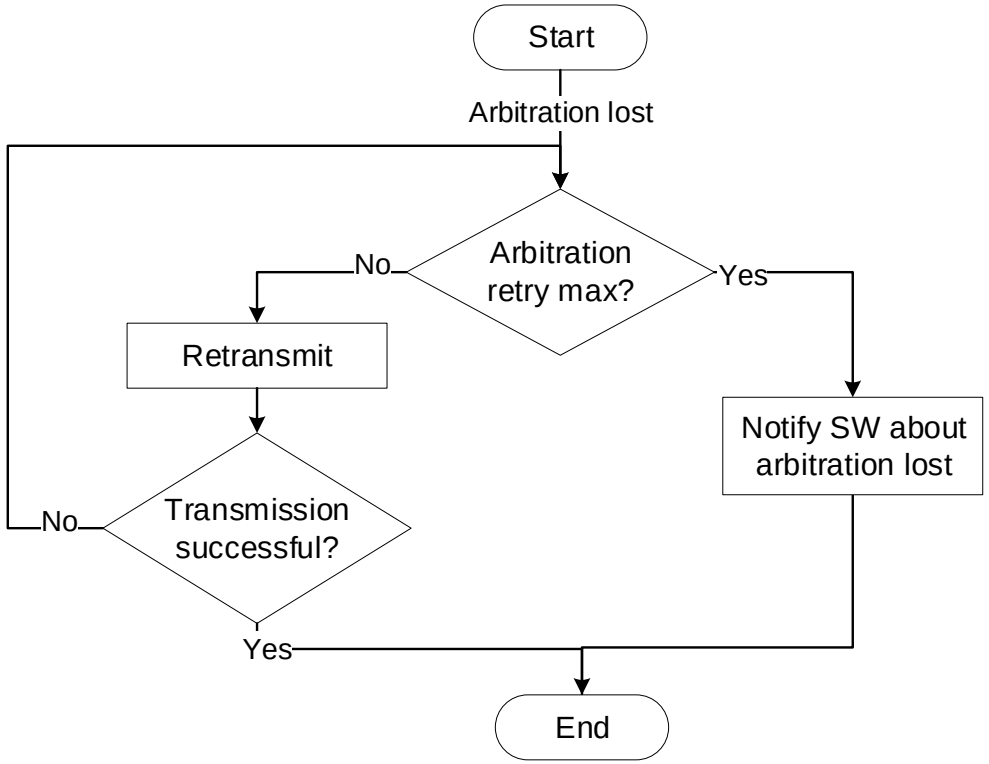
Hint Bar

Review TRM section 30.5.1 for additional details

Arbitration takes place according to PID field priority. The smaller frame ID with the bit of logical value '0' has priority over the bit of logical value '1'

Arbitration Loss

- Defined as TX and RX bit mismatch during transmission of the PID field or PTYPE field1
- If an arbitration is lost, HW checks whether it has reached its maximum amount of retries via CTL2.RETRY



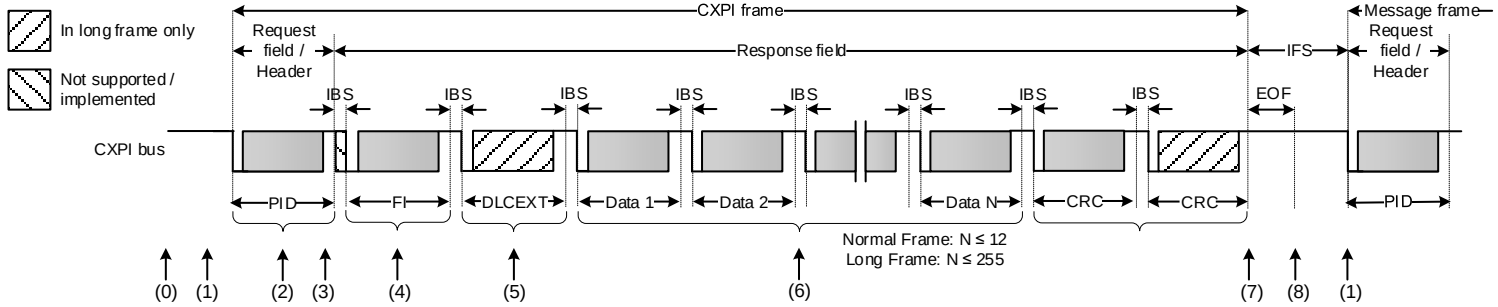
[1] Excluding Start bit or Stop bit

Hint Bar

Review TRM section 30.5.9 and Register TRM for additional details

Event Trigger Method Operation (1/2)

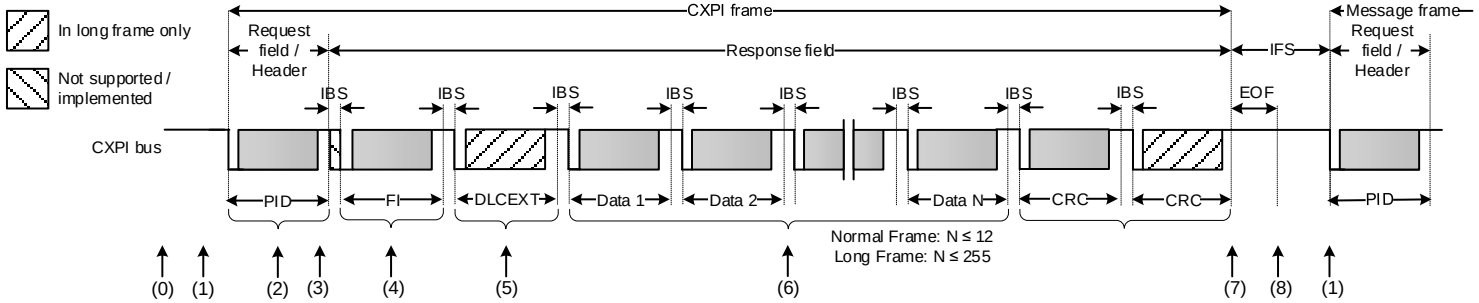
› Transmission of PID and RESPONSE by Master/Slave



Step	Software Processing	CXPI Hardware Processing
(0)	1. Channel initialization: • Set master/slave and modulation mode (NRZ/PWM) • Automatic Transceiver Handling ON/OFF • Receive PID Zero Check OFF (only for slave in Polling method) • RX filtering ON/OFF • Set IFS length and IBS length • TX abort for bit error detection ON/OFF • Define PWM pulses (only PWM mode) • Configure RX sample point • Define TX wake-up pulse length • Select Time-out and configure FIFO 2. Start: • Enable Channel and transit CXPI bus to Normal mode	1. Channel disabled 2. Entering normal mode
(1)	• PID field Transmission (master/ slave) • Write PID value • Set CMD.WAIT_IFS • Set CMD.RX_RESPONSE • Set CMD.TX_HEADER (trigger PID transmission) • PID field Reception (master/ slave): • Set CMD.RX_RESPONSE • Set CMD.RX_HEADER	Waiting for transfer

Event Trigger Method Operation (2/2)

› Transmission of PID and RESPONSE by Master/Slave



Step	Software Processing		CXPI Hardware Processing	
(2)	-		Transferring PID	
(3.1)	Process PID (due to possible arbitration)		INTR.RX_HEADER_PID_DONE flag set	

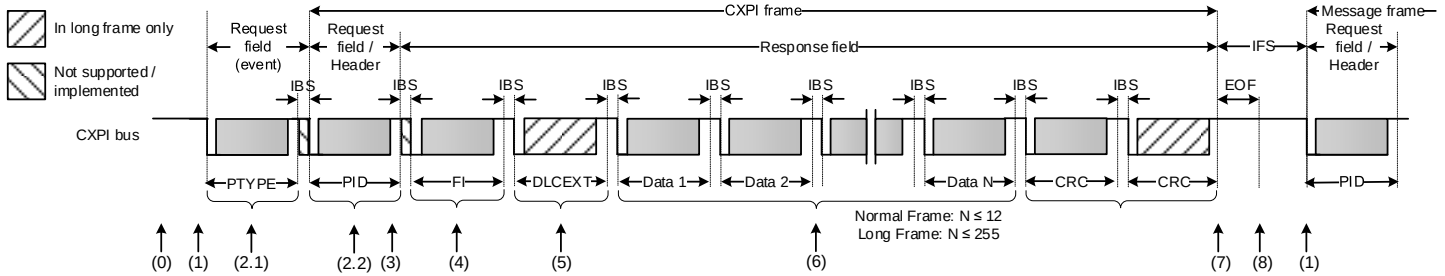
(3.2)	• Response Transmission:
	  • Write FI value
	  • Write DLEXT value[1]
	  • Write data to TX FIFO
	  • Clear CMD.RX_RESPONSE
	  • Set CMD.TX_RESPONSE
		• Response Reception:
		  • No action
			• Response Transmission:
			  • Waiting for TX request
				• Response Reception

(4)	-	-	• Transmitting FI field IBS transmitter	• Receiving FI field received
(5)	-	-	• Transmitting DLCEXT field and IBS[1]	• DLCEXT field received[1]
(6.1)	• Process TX FIFO[1]	• Process RX FIFO[1]	• Transmitting data bytes and IBS	• Receiving data bytes
(6.2)	-		• Data field transmission completed • Transmitting CRC	• Data field reception completed • Receiving CRC
(7)	Frame post processing		• Response succeeded	• Response succeeded
(8)	-		End-of-Frame (EOF) 10 Tbit after response completion INTR.TXRX_COMPLETE flag is set	

[1] In long frame only

Polling Method Operation (1/2)

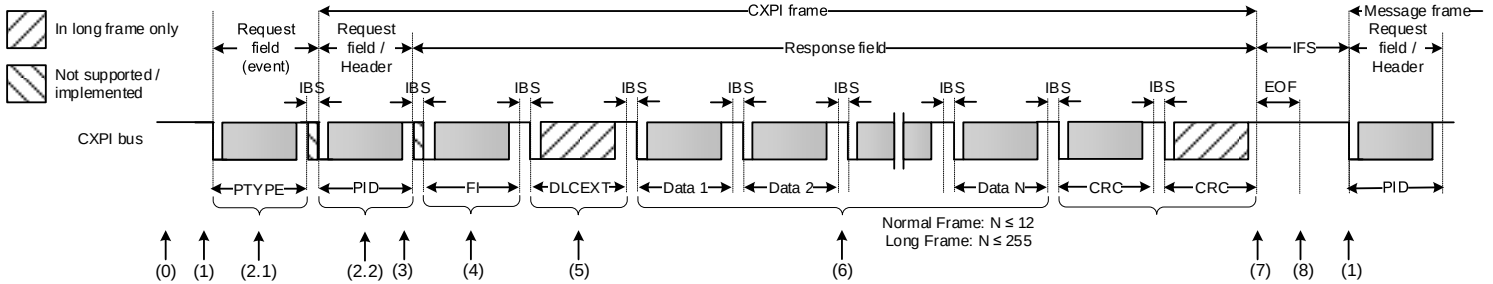
› Transmission of PID and RESPONSE by Master/Slave, and transmission of PTYPE by Master



Step	Software Processing		CXPI Hardware Processing
(0)	1. Channel initialization: • Set master/slave and modulation mode (NRZ/PWM) • Automatic Transceiver Handling ON/OFF • Receive PID Zero Check ON (only for slave in Polling method) • RX filtering ON/OFF • Set IFS length and IBS length • TX abort for bit error detection ON/OFF • Define PWM pulses (only PWM mode) • Configure RX sample point • Define TX wake-up pulse length • Select Time-out and configure FIFO 2. Start: • Enable Channel and transit CXPI bus to normal mode		1. Channel disabled 2. Entering normal mode
(1)	1. Master only: • Set CMD.WAIT_IFS • Write PID/PYTPE value • Set CMD.RX_RESPONSE • Set CMD.RX_RESPONSE • Set CMD.TX_HEADER (trigger PID transmission)	1. Slave only: • Write PID value (PTYPE case only) • Set CMD.RX_RESPONSE • Set CMD.RX_HEADER • Set CMD.TX_HEADER	Waiting for transfer

Polling Method Operation (2/2)

› Transmission of PID and RESPONSE by Master/Slave, and transmission of PTYPE by Master

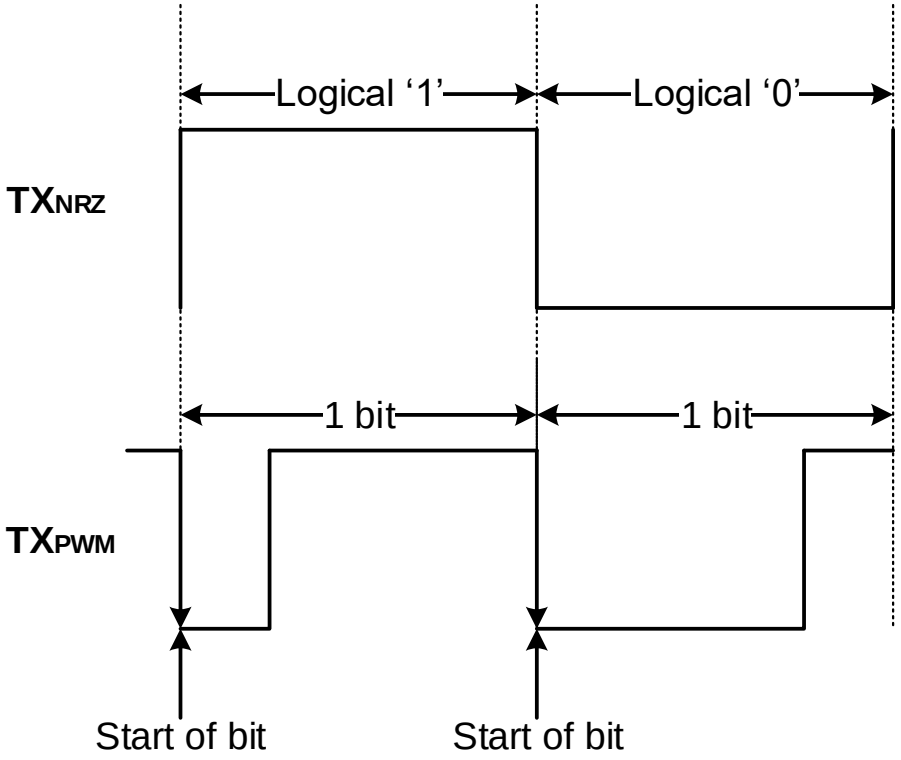


Step	Software Processing		CXPI Hardware Processing	
(2.1)	-	-	Transferring PTYPE (optional)	
(2.2)	-	-	Transferring PID	
(3.1)	Process PID (check for arbitration loss (slave only))		INTR.RX_HEADER_PID_DONE flag set	
(3.2)	1. Response Transmission: • Write FI value • Write DLEXT value[1] • Write data to TX FIFO • Clear CMD.RX_RESPONSE • Set CMD.TX_RESPONSE	1. Response Reception: • No action	• Response Transmission: Waiting for TX request	• Response Reception
(4)	-	-	• Transmitting FI field and IBS transmitted	• Receiving FI field received
(5)	-	-	• Transmitting DLCEXT field[1] and IBS	• DLCEXT field received[1]
(6.1)	• Process TX FIFO[1]	• Process RX FIFO[1]	• Transmitting data bytes and IBS	• Receiving data bytes
(6.2)	-		• Data field transmission completed • Transmitting CRC	• Data field reception completed • Receiving CRC
(7)	Frame post processing		• Response succeeded	• Response succeeded
(8)	-		EOF 10 Tbit after response completion INTR.TXRX_COMPLETE flag is set	

[1] In long frame only

Bus Signal Modulation (1/3)

- The CXPI channel can process NRZ signals (CTL0.MODE = 0) and PWM signals (CTL0.MODE = 1)
- The following diagram shows the encoded CXPI bus signals:



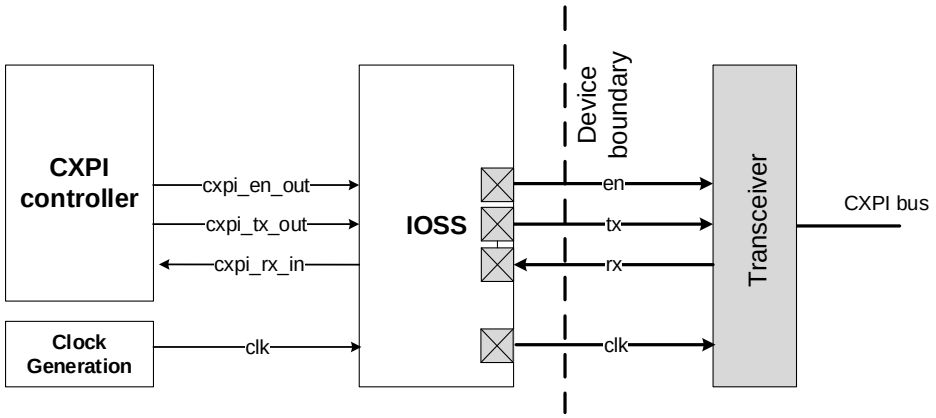
Hint Bar

Review TRM section 30.5 for additional details

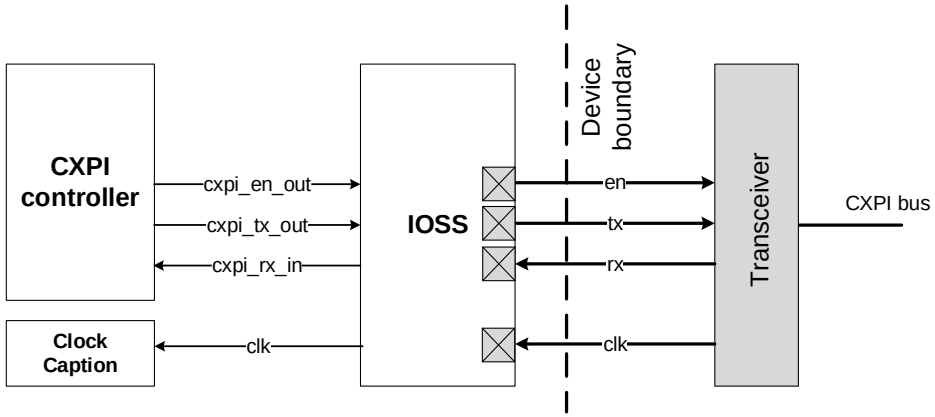
Bus Signal Modulation (2/3)

- Non-Return to Zero (NRZ) mode
  - Since the channel does not provide the CXPI clock signal, the clock must be generated by another module separately

Master in NRZ mode

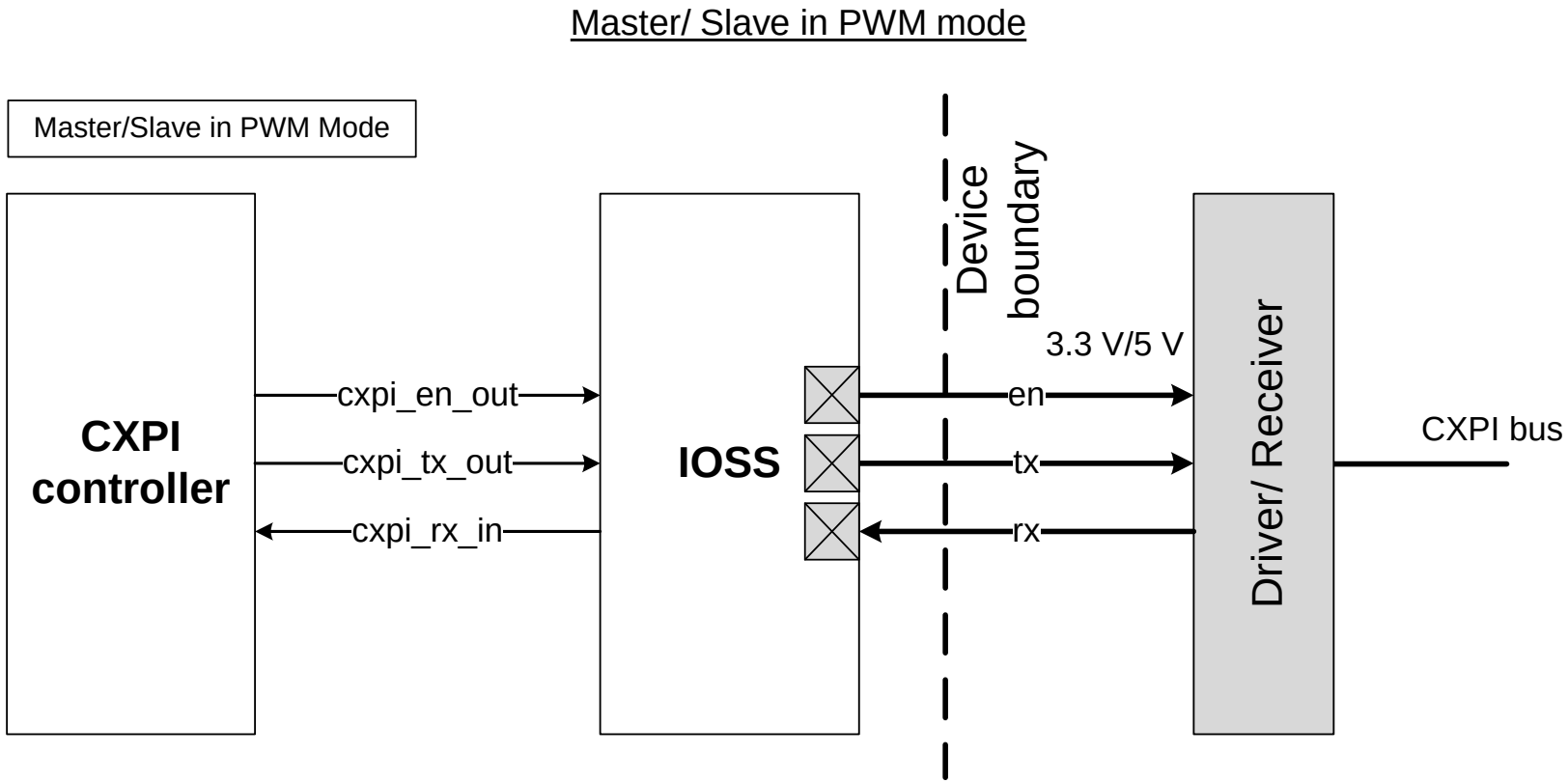


Slave in NRZ mode



Bus Signal Modulation (3/3)

- Pulse-Width Modulation (PWM) mode
- PWM encoding and decoding is done in the CXPI channel
- Additional device is not needed to generate the clock on the CXPI bus



Hint Bar

Review TRM section 30.5 for additional details

FIFO Buffer

- Every channel has two separate TX and RX FIFO buffers with 16-byte depth each[1]
- Provides “underflow” and “overflow” events
  - An “underflow” event is triggered by an attempt to read from an empty FIFO
  - An “overflow” event is triggered when the FIFO data is overwritten

Hint Bar

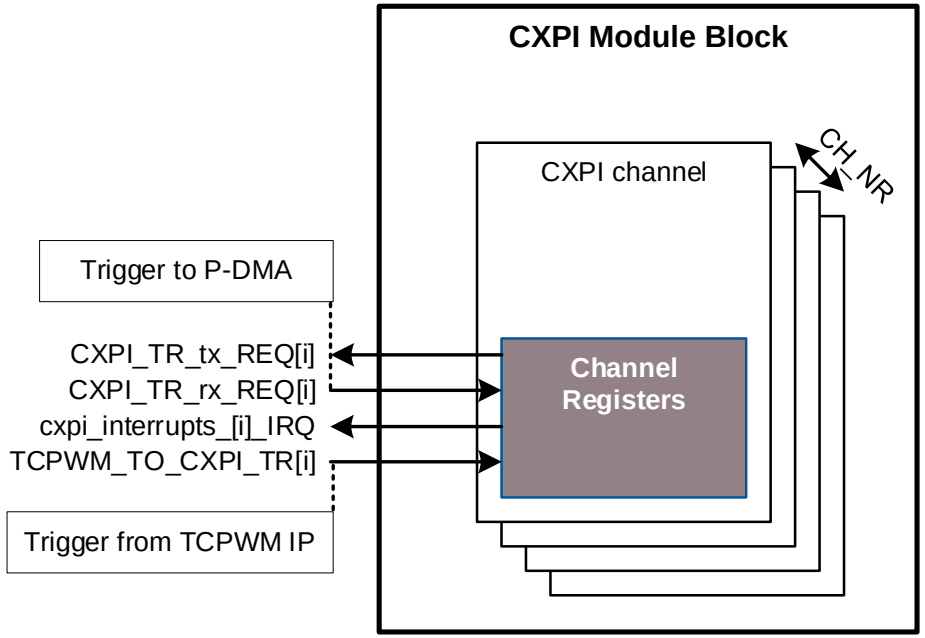
Review TRM section 30.5.7 for additional details

FIFO buffers are only applied in the Data Field

[1] Refer to the Register TRM (INTR.TX, INTR.RX) for additional details.

P-DMA Transfer Trigger

- To avoid additional CPU access to the FIFO buffers, every CXPI channel is connected to P-DMA with trigger signal lines for both FIFO buffers
- P-DMA trigger is set in the following instances:
  - Transmission:
    - TX_FIFO_STATUS.USED < TX_FIFO_CTL.TRIGGER_LEVEL
  - Reception:
    - RX_FIFO_STATUS.USED > RX_FIFO_CTL.TRIGGER_LEVEL



Hint Bar

Review TRM section 30.5.7 for additional details

Timeout Detection

- The timeout feature has the following configurations:

Timeout Detection Selection (TIMEOUT_SEL)	Timeout Detection: header-header	Timeout Detection: header-response	Timeout Detection: header-header-response
0 (OFF)	No	No	No
1	No	Yes	No
2	Yes	Yes	Yes

- Timeout length[1] is configured in Tbit
- TIMEOUT_SEL=1: 2 Tbits
- TIMEOUT_SEL=2: 3 Tbits

Hint Bar

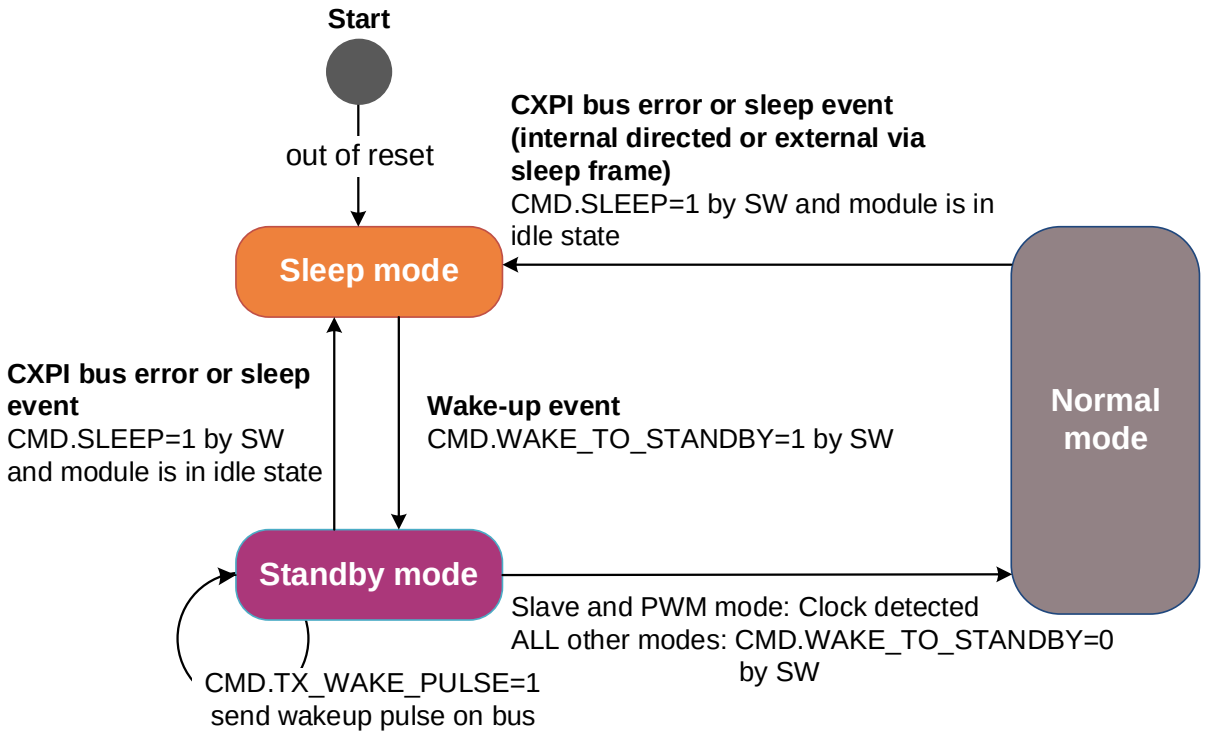
Review TRM section 30.7.1 and Register TRM for additional details

CXPI channel uses a fixed oversampling of 400, meaning 1 Tbit is equivalent to 400 samplings

[1] You must not set IBS>TIME_LENGTH.

Power Modes

› The timeout feature has the following configurations:



Command	Description
WAKE_TO_STANDBY	Direct HW to wake up from Sleep mode to Standby mode
TX_WAKE_PULSE	Direct HW to send wake-up pulse
SLEEP	Direct HW to sleep mode

Hint Bar

Review TRM section 30.5.3 and Register TRM for additional details

These power modes are different from the device power save modes of the device

Enabling CXPI Channel

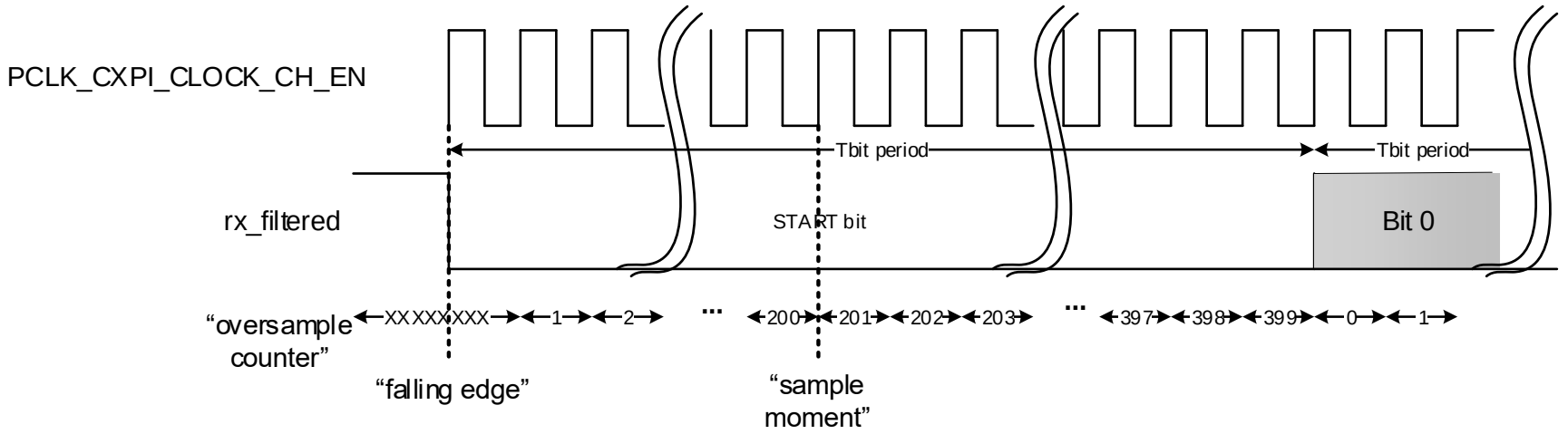
- Each CXPI channel must be enabled by the CTL0.ENABLED bit
- When the same bit is cleared, all registers are cleared except the control registers
- After it is re-enabled, the CXPI channel restarts from the Sleep mode

Hint Bar

Review TRM section 30.3 and Register TRM for additional details

Oversampling

- One CXPI bit length corresponds to 400 PCLK_CXPI_CLOCK_CH_EN[1] cycles
- The sample moment is the sampling point at which the value of the received signal is detected for further channel processing
- The sample moment can be configured in CTL1.T_OFFSET



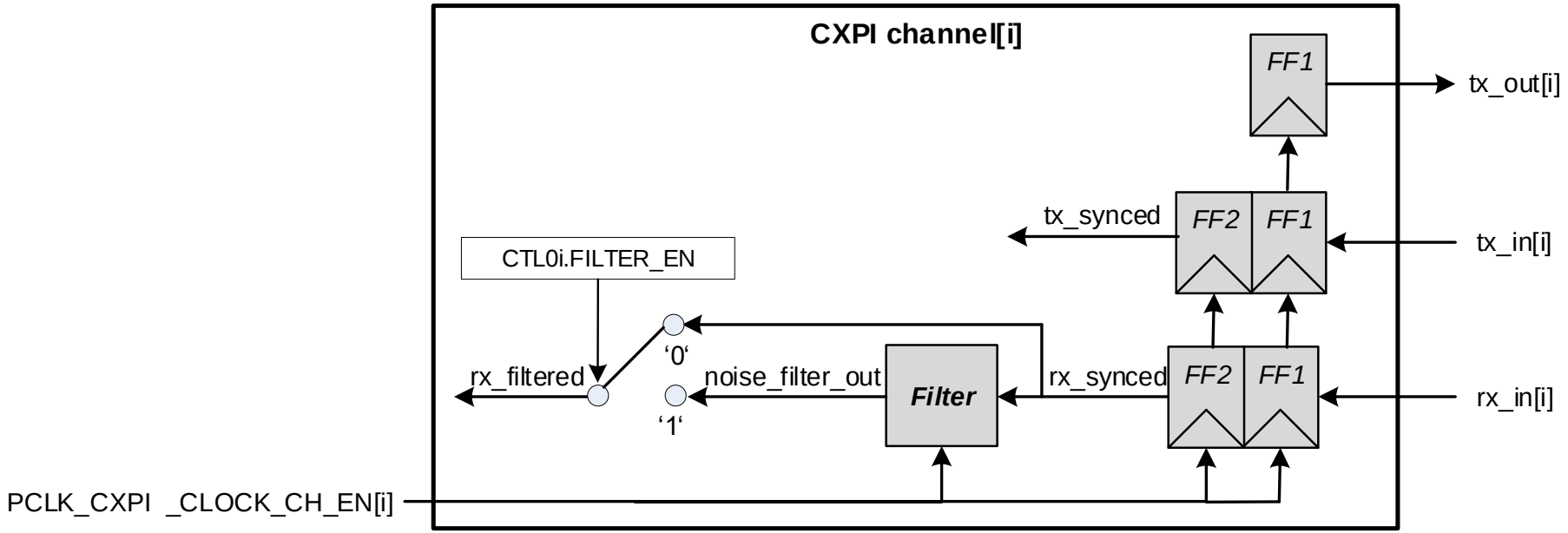
Hint Bar

Review TRM section 30.3 and Register TRM for additional details

[1] A dedicated internal CXPI channel clock derived from the clk_peri, which would dictate the baud rate of the CXPI functionality.

Noise Filter

- If CTL0.FILTER_EN = 1, “rx_synced” is fed to a three-input median filter and outputs the result as “rx_filtered”

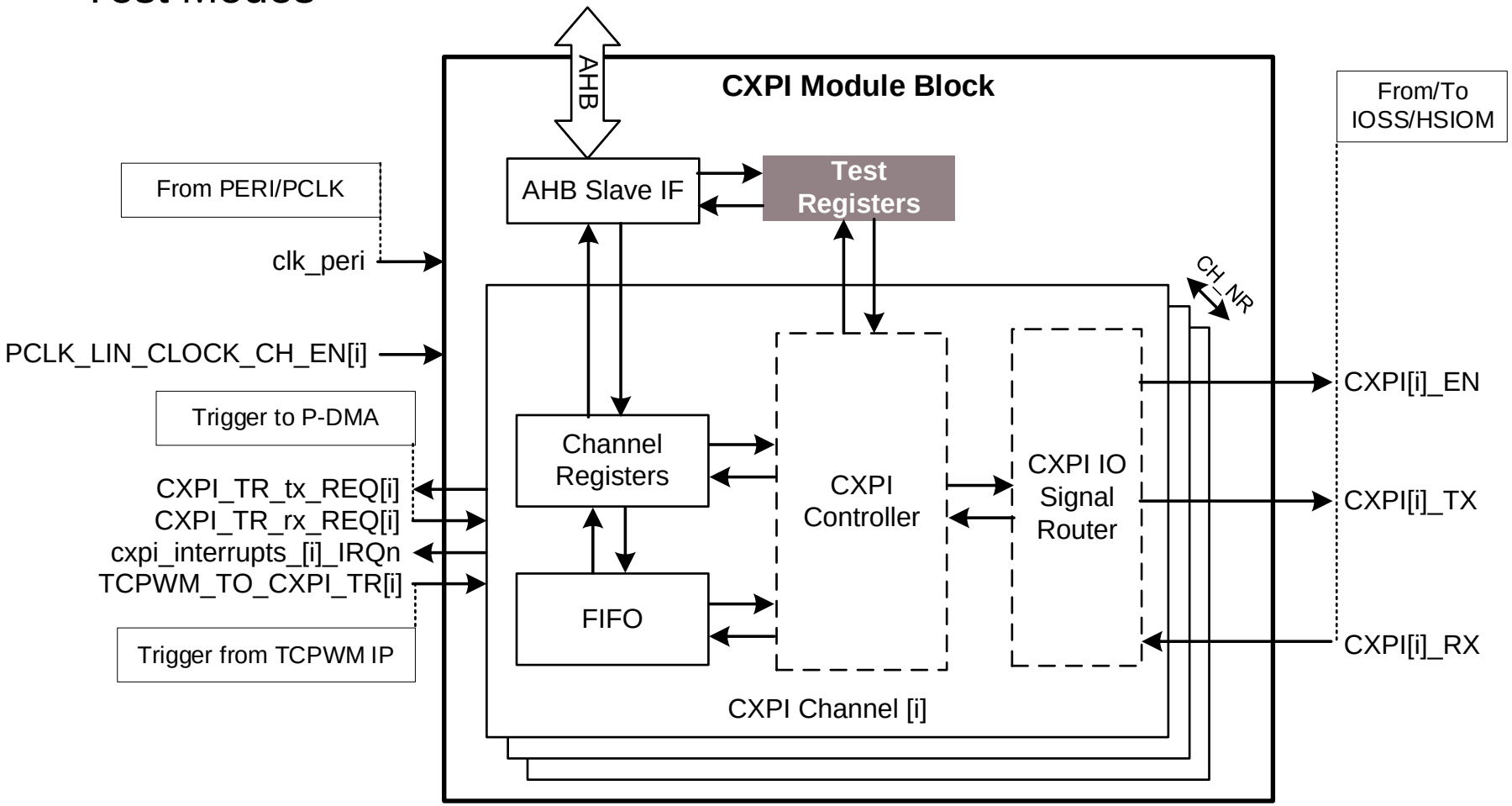


Hint Bar

Review TRM section 30.3.3 for additional details

CXPI Module Block Components

- CXPI Test Registers
  - Test Modes

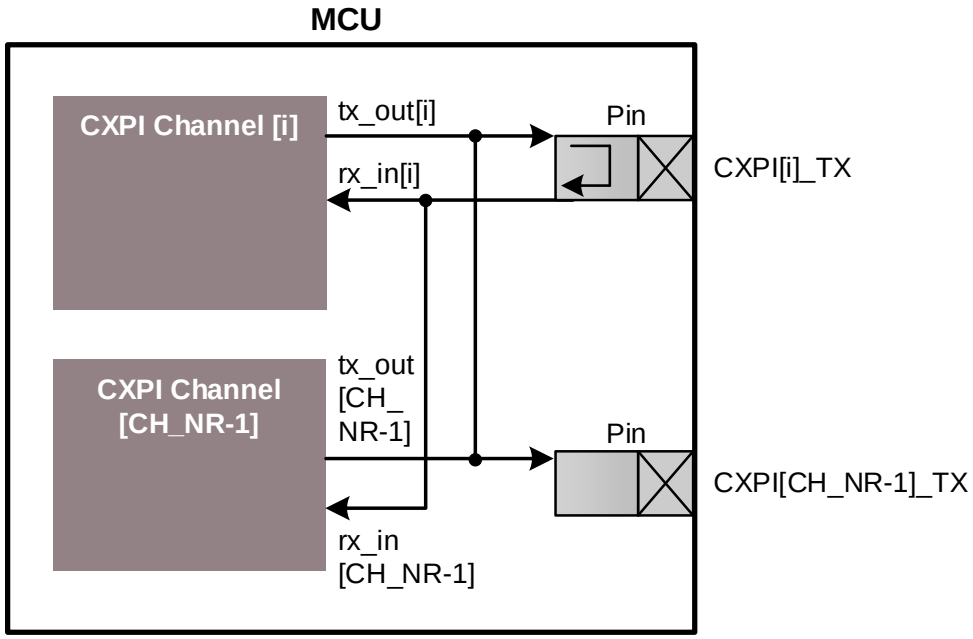


Hint Bar

Review TRM section 30.6 for additional details

Test Modes (1/2)

- Partial Disconnect Mode
  - Loopback mode is done via the IOSS port pin structure
  - Connection between CXPI ch.[i] and CXPI ch.[CH_NR-1]



- Advantage
  - The communication can be monitored outside the device

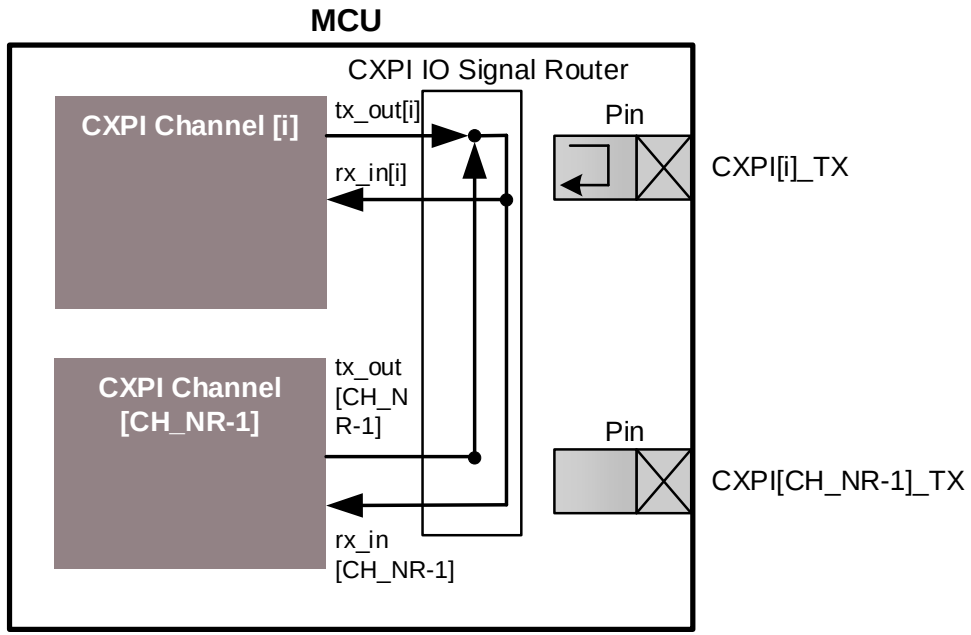
Hint Bar

Review TRM section 30.6.2 for additional details

Max channel number (CH_NR)

Test Modes (2/2)

- Full Disconnect Mode
  - Full Loopback mode between CXPI ch.[i] and CXPI ch.[CH_NR-1]
  - There is no connection to port pins



- Advantage
  - Test can be performed without any external transceivers or device involvement

Hint Bar

Review TRM section 30.6.2 for additional details

Part of your life. Part of tomorrow.

002-25755 *B, 2020-12-16

Revision History

Revision	ECN	Submission Date	Description of Change
**	6401064	12/04/2018	Initial release
*A	6649384	08/07/2019	Added the note descriptions of all pages. Added slides 4, 16 and 18. Updated page 2, 8, 10, 11, 13, 15, 17 and 25.
*B	7053066	12/16/2020	Updated page 3, 22