- DISCONTINUED PRODUCT -

XPS FlexRay Controller (v1.00a)

Introduction

The Xilinx FlexRay LogiCORE product specification defines the architecture and features of the Xilinx FlexRay controller. This document also defines the functionality of the modules in the design, in addition to defining the register addressing of the Xilinx FlexRay controller. The scope of this document does not extend to describing the FlexRay protocol and assumes knowledge of the specifications described in the Reference Documents section.

Features

- Conformance with FlexRay protocol specification version 2.1 rev A.
- Industrial (I) , Extended Temperature Range (Q) and Automotive grade device support.
- Single Communication Channel.
- Data rate upto 10 Mbps.
- Scalable synchronous and asynchronous data transmission.
- Configurable Payload length upto a maximum of 256 bytes.
- Configurable Tx Buffers for storage of upto 128 messages.
- Configurable Rx Fixed Buffers for storage of upto 128 messages.
- Configurable Rx FIFO buffer for storage of upto 128 messages.
- Frame ID, Cycle Counter and Message ID based receive filtering.
- 32 bit PLB interface with single read/write support.
- Variable PLB interface clock.

LogiCORE™ Facts		
Core Specifics		
Supported Device Family	Spartan™-3, Spartan-3XA, Spartan-3A, Spartan-3A DSP, Spartan-3E, Spartan-3 EXA, Virtex™-II PRO, Virtex-4	
Version of core	xps_flexray	v1.00a
Resources Used		
	Min	Max
Slices	Refer to Table 231	
LUTs		
FFs		
Block RAMs		
Special Features	None	
Provided with Core		
Documentation	Product Specification	
Design File Formats	VHDL	
Constraints File	UCF	
Verification	VHDL Test bench	
Instantiation Template	VHDL Wrapper	
Reference Designs & application notes	None	
Additional Items	None	
Design Tool Requirements		
Xilinx Implementation Tools	ISE (version) or later	
Verification	ModelSim (version)	
Simulation	ModelSim (version)	
Synthesis	XST	
Support		
Provided by Xilinx, Inc.		

Functional Description

Figure 1 shows the high level architecture of the Xilinx FlexRay controller.

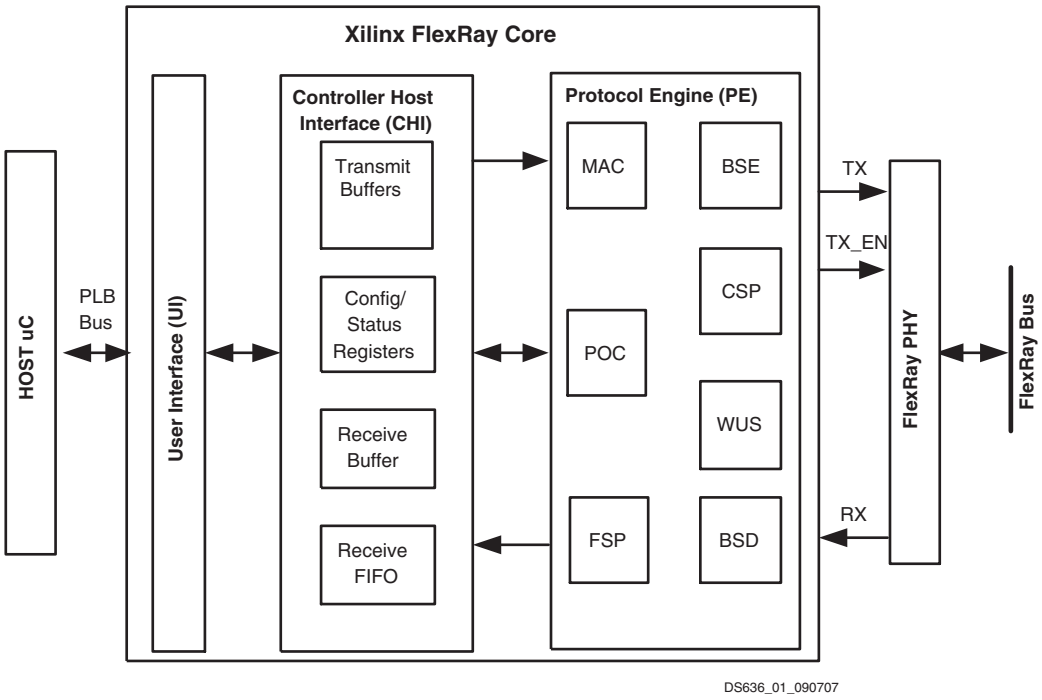

Figure 1: **Xilinx FlexRay Controller Architecture**

The Xilinx FlexRay controller contains three primary functional units/modules:

1. User Interface (UI)
2. Controller Host Interface (CHI)
3. Protocol Engine (PE)

User Interface (UI)

This primary function of UI module is to provide PLB bus connectivity to the FlexRay controller. The UI module performs the following tasks:

- PLB Read/Write transactions
- Interrupt management

PLB Read/Write transactions

Refer Timing Diagrams section (Figure 3, Figure 4, Figure 5, Figure 6) of this document for more details on the PLB Single Read/Write and Burst Read/Write transactions.

Interrupts management

The controller provides four active-high interrupt lines to alert the host of the occurrence of interrupt events. The following are the active-high interrupt lines:

- Flex_Cirpt: Indicates interrupt conditions due to error conditions, timers and change in operational state of the FlexRay controller.
- Flex_Txbirpt: Indicates interrupt conditions associated with the Transmit Buffers.
- Flex_Rxbirpt: Indicates interrupt events associated with the Rx Buffers.
- Flex_Rxfirpt: Indicates interrupt events associated with the Rx FIFO.

Controller Host Interface (CHI)

The Controller Host Interface manages message, control and configuration data flow between the host processor and the Protocol Engine (PE).

The CHI contains 4 major blocks:

1. Memory Map and Registers

2. Transmit buffers

3. Receive buffers

4. Receive FIFO

Memory Map and Registers

This module contains the control, status and configuration memory map and registers. It allows read and write access to control the following register sets:

- Protocol configuration, control, and status
- General CHI configuration, control, and status

Transmit buffers

The Transmit buffers provide storage and control of message data to be transmitted over the FlexRay physical interface. The Xilinx FlexRay controller provides the user, a configurable number of upto 128 Transmit buffers, each of which can store one FlexRay frame of variable payload length. The number of Transmit buffers in the controller can be configured by the generic: `Number of Tx Buffers`.

The maximum number of Transmit buffers to which the controller can be configured to contain, is dependent on the generic: `Max Payload Size`. This number can vary from a minimum of 32 buffers (Maximum Payload Size set to 256) to a maximum of 128 Transmit buffers (Maximum Payload Size set between 2 and 64).

Receive buffers

Receive buffers provide storage for received message data. Receive buffers perform the following functions:

1. Frame storage

2. Frame filtering

The Xilinx FlexRay controller provides a user configurable number, upto 128, of Receive buffers, each of which can store one FlexRay frame of variable payload length. The number of Receive buffers in the controller can be configured using the generic `Number of Rx Buffers`.

Filtering

Each Receive Buffer is associated with a filter which can be programmed to accept a particular combination of Frame ID, Cycle counter, and Message ID values for storage into the buffer.

Receive FIFO

In addition to Receive buffers described above, the Xilinx FlexRay controller provides a flexible, variable length FIFO buffer for storage of received FlexRay frames. The message storage space allocated to the Receive FIFO can be configured by the user by setting the generic `Depth of Rx FIFO`.

Filtering

Four acceptance filter pairs are provided to screen incoming messages from the Protocol Engine for storage into the Rx FIFO. Each filter pair contains a mask and data pair, each of which in turn contains fields for Frame ID, Cycle counter and Message ID.

Protocol Engine(PE)

The Protocol Engine module contains all functional blocks which perform low-level FlexRay protocol related tasks described in the protocol specification. These sub blocks are:

- Media Access Control
- Bit Stream Encoder
- Bit Stream Decoder
- Frame and Symbol Processing
- Protocol Operation Control
- Wakeup and Startup
- Clock Synchronization Process

These sub-blocks are described below:

Media Access Control (MAC)

The MAC is used to maintain the timing within each communication cycle and to manage the timing of static and dynamic slot duration. It also manages the transmit operation of the node and handles frame transmission by assembling the frames.

MAC maintains timing consistency within a communication cycle for the following segments:

- Static
- Dynamic
- Symbol Window
- Network Idle Time

The MAC asserts signals to other modules to indicate the static segment start, slot boundary, dynamic segment start, symbol window start and network idle start time.

Bit Stream Encoder (BSE)

This module handles encoding of frames and symbols. Both Frame and Symbol encoding are described below.

Frame Encoding

The encoder takes the incoming bytes from the MAC and assembles the FlexRay bit stream by:

- Appending a Transmit Start Sequence (TSS) to start of bit stream
- Appending a Frame Start Sequence (FSS) immediately after the TSS
- Adding a Byte Start Sequence (BSS) at the beginning of each byte
- Transmitting each byte serially
- Calculating Frame CRC and appending it to the end of the Frame
- Appending a Frame End Sequence (FES)
- Appending a Dynamic Trailing Sequence (DTS) if the frame is transmitted in dynamic segment

Symbol Encoding

Three types of symbols can be transmitted – two of which are identical in terms of bit pattern. They are:

- CAS – Collision Avoidance Symbol
- MTS – Media Test Symbol.
- WUS – Wakeup Symbol

Bit Stream Decoder (BSD)

The Decoding block performs the following functions:

- Sampling and Majority Voting
- Bit Clock Alignments (BCA) and Bit Strobing
- Frame and Symbol Decoding
- Channel Idle detection
- Decoding Error detection

This Frame and Symbol Decoding modules detect various signaling patterns, which can be broken down into the following sections:

- Communication Element CE which indicates the active period of a symbol/frame
- Channel idle detection which operates closely with CE generation
- TSS detection (common to frames and symbols)
- Symbol detection
- Frame detection
- Time Reference point generation
- Decoding Error detection

Frame detection consists of the following functions

- Checking all BSS indicators during the frame
- FES detection
- Control of Header CRC check
- Control of Trailer segment CRC check

Frame and Symbol Processing (FSP)

Frame and symbol processing checks the correct timing of frames and symbols with respect to the TDMA scheme, applies further syntactical tests to received frames and checks the semantic correctness

of received frames. The result of these checks are signaled to Host. The following status is provided by the FSP to the Host.

- Valid Frame
- Valid Symbol
- Syntax Error
- Content Error
- Boundary Violation
- Tx Conflict

Frame detection consists of the following functions

- Checking all BSS indicators during the frame
- FES detection
- Header CRC checking.
- Trailer segment CRC checking

Protocol Operation Control (POC)

The POC acts as an interface between the Host and FlexRay PE sub-modules. The POC determines the operational state of the controller.

POC state transitions are controlled synchronously and occur as a consequence of a host command or the occurrance of an error condition. The state transitions cause a change in operation of the protocol engine.

Refer Figure 2 for POC state machine.

Wakeup and Startup (WUS)

The Wakeup procedure describes the transition of a node from sleep mode to operational mode and the Startup procedure describes the integration of nodes into the cluster. The Controller Host Interface triggers both the procedures.

The wakeup pattern transmitted consists of a programmable number of wakeup symbols. If more than one node tries to wakeup a channel, the wakeup procedure resolves the situation such that only one node ends up transmitting the pattern. The outcome of the wakeup operation is readable by the host.

Wake-up Block

The wakeup procedure is triggered from a host command. The state machine first enters a wakeup listen state where it remains for the duration defined by the timer in the Listen Timeout register. If no communication is observed during this time-frame, then the send state is entered and the wakeup pattern is transmitted. If communication is observed while the wakeup pattern is being sent, then the wakeup detect state is entered and the cause of the collision is investigated.

Startup Block

The process by which startup occurs is different for leading cold start nodes, integrating coldstart nodes and integrating nodes. The following are the major Startup process types handled by the Startup module:

- Leading cold start node startup

- Integrating cold start node startup
- Integrating non cold start node startup

Clock Synchronization Process (CSP)

The Clock Synchronization Process performs the synchronization of all nodes in a cluster to ensure that they share the same global view of time. Synchronization is achieved by using sync frames, which are sent by nodes on the network. There are two components of time adjustment that are handled by the CSP: Rate correction and Offset correction.

The two major sub-processes of the CSP that perform the above functions are:

- Clock Sync Processing
- Clock Sync Startup

Clocking and Reset

Clocking

The Xilinx FlexRay controller uses three clocks. System clock, Sample clock, and Interface clock

- System clock

The FlexRay controller uses a System clock of 40 MHz. The System clock must be generated by a DCM. The input to DCM can be from an external oscillator and in such a case the external oscillator must meet the DCM input clock requirements. The PE module runs off the System clock.

- Sample clock

The Sample clock has a frequency of 80 MHz. The Sample clock can be derived from the same DCM that generates the System clock or it can be derived from another DCM with the System clock as the input to the DCM. The Sample clock is used to perform the oversampling of the incoming receive data. The Sample clock is used by the PE module.

- Interface clock

This is the PLB bus clock. The user can specify the operating frequency of PLB clock. Using a DCM to generate the Interface clock is optional. The UI and the CHI modules use the Interface clock.

Reset

Two different reset mechanisms are provided for the FlexRay controller. The PLB_Rst input acts as the System reset. Apart from the System reset, a Software reset is provided through the SRST bit in the SRR.

Software Reset

The Software reset is asserted by writing a '1' to the SRST bit in the Software Reset register (SRR). The assertion of Software reset places the FlexRay controller in the Reset state. Upon completion of the Reset sequence, the FlexRay controller transitions to the Default Config state. In the Reset state, reads to the SRST bit return a '1'. The application of Software reset during normal operation of the FlexRay controller immediately places the node into Reset state and current frame transmission or reception is abruptly stopped. The Software Reset does not clear the contents of the Block RAMs used in the FlexRay controller. The Software reset resets the CHI and PE modules.

System Reset

The System reset can be enabled by driving a '1' on the PLB_Rst input. Upon deassertion of the System reset, the controller enters the Reset state. Upon completion of the Reset sequence, the FlexRay controller transitions to the Default Config state. As long as the System reset is asserted, reads and writes to the FlexRay controller memory map are prohibited. The application of System reset during normal operation of the FlexRay controller immediately places the controller into Reset state and current frame transmission or reception is abruptly stopped. The System Reset does not clear the contents of the Block RAMs used in the FlexRay controller. Assertion of the System reset resets the UI, CHI and the PE modules, and upon deassertion of the System reset, the UI module is no longer in reset.

Interrupts

The FlexRay IP Core uses a hard-vector interrupt mechanism. Each of the four interrupt lines (`Flex_Cirpt`, `Flex_Txbirpt`, `Flex_Rxbirpt`, `Flex_Rxfirpt`) indicates an interrupt on transition from a logic '0' to a logic '1.' During power up, the Interrupt line is driven low.

The Interrupt Status Registers (`CISR`, `TXBISR[n]`, `RXBISR[n]`, `RXFISR`) indicate the interrupt status bits. These bits are set and cleared regardless of the status of the corresponding bit in the Interrupt Enable Registers (`CIER`, `TXBIER[n]`, `RXBIER[n]`, `RXFIER`). The Interrupt Enable Registers handle the interrupt-enable functionality. Clearing a status bit in the Interrupt Enable Registers is handled by writing a '1' to the corresponding bit in the Interrupt Clear Registers (`CICR`, `TXBICR[n]`, `RXBICR[n]`, `RXFICR`).

The following conditions cause the interrupt lines to be asserted:

- If a bit in the Interrupt Status Register is '1' and the corresponding bit in the Interrupt Enable Register is '1.'
- An Interrupt Enable Register bit from a '0' to '1' is changed when the corresponding bit in the Interrupt Status Register is already '1.'

Two conditions cause the interrupt lines to be deasserted:

- Writing a '1' to a bit in the Interrupt Clear Register and the corresponding bit in the Interrupt Enable Register is '1.'
- Changing an Interrupt Enable Register bit from '1' to '0'; when the corresponding bit in the Interrupt Status Register is '1.'

When both deassertion and assertion conditions occur simultaneously, the interrupt line is deasserted first, and is reasserted if the assertion condition remains true.

Protocol Operational States

The FlexRay controller supports the following operational states:

- Reset state
- Default Config state
- Config state
- Ready state
- Wakeup state
- Startup state
- Normal Active state
- Normal Passive state

- Halt state

The operational state transitions occur because of the reset conditions, Host commands and internal protocol error conditions. Resets are asserted by writing to the SRR and by asserting the PLB_Rst line. The host can issue commands by writing to the CHI Command register.

Reset state

In the Reset state:

- The CHI and PE modules are reset. It should be noted that when the PLB_Rst signal is asserted the UI module is Reset. Upon deassertion of the PLB_Rst signal the UI module is no longer reset.
- The FlexRay controller registers are reset to their default values. However registers and memory locations that are stored in BRAMs are not reset.
- Read access to the FlexRay controller Memory map is permitted.

The FlexRay controller enters the Reset state as follows:

- The system reset is asserted by driving a '1' on the PLB_Rst input. The controller continues to stay in Reset state as long as the PLB_Rst input is '1'.
- The software reset is asserted by writing a '1' to the SRST bit in the SRR.

The FlexRay controller transitions to the Default Config state from the Reset state when the Reset sequence is completed.

Default Config state

In the Default Config state:

- The PE module is disabled.

The FlexRay controller enters the Default Config state as follows:

- The controller receives the CHI_DEFAULT_CONFIG command (from the Halt state)
- The Reset sequence is completed (from the Reset state).

The FlexRay controller transitions from the Default Config state as follows:

- The controller receives the CHI_CONFIG command (to the Config state)
- The assertion of Software reset or Hardware reset (to the Reset state).

Config state

In the Config state:

- Configuration of the Protocol Registers, Transmit Buffers and Receive Buffers is to be performed.
- The PE module is disabled.

The FlexRay controller enters the Config state as follows:

- The controller receives the CHI_CONFIG command (from the Default Config state).
- The controller receives the CHI_CONFIG command (from the Ready state).

The FlexRay controller transitions from the Config state as follows:

- The controller receives the CHI_CONFIG_COMPLETE command (to the Ready state).

- The assertion of Software reset or Hardware reset (to the Reset state).

Ready state

In the Ready state:

- The PE module is enabled.
- Protocol specific configuration registers should not be written to.

The FlexRay controller enters Ready state when any of the following conditions occur:

- It receives the CONFIG_COMPLETE command from the host (from the Config state).
- It receives the READY command from the host (from the Normal Active state, Normal Passive state, Wakeup state, Startup state).

The FlexRay controller exits the Ready state when the following condition occurs:

- When the controller receives the WAKEUP command from the host (to the Wakeup state).
- When the controller receives the CHI_RUN command (to the Startup state).
- When the controller receives the CHI_CONFIG command (to the Config state).

Wakeup State

In the Wakeup state

- The node performs the wakeup operation

The FlexRay controller enters the Wakeup state as follows:

- The controller receives the CHI_WAKEUP command (from the Ready state).

The FlexRay controller transitions from the Wakeup state as follows:

- The controller receives the CHI_FREEZE command (to the Halt state)
- The controller receives the CHI_READY command (to the Ready state)
- The completion of the wakeup operation (to the Ready state).
- The assertion of Software reset or Hardware reset (to the Reset state).

Startup state

In the Startup state

- FlexRay controller performs the startup operation based on the mode of the node. The mode for a particular node can be coldstart or integrating node

The FlexRay controller enters the Startup state as follows:

- The controller receives the CHI_RUN command (from the Ready state).

The FlexRay controller transitions from the Startup state as follows:

- The controller receives the CHI_FREEZE command (to the Halt state)
- The controller receives the CHI_READY command (to the Ready state)
- The completion of the startup operation (to the Normal Active state).
- The assertion of Software reset or Hardware reset (to the Reset state).

Normal Active state

In the Normal Active state:

- The FlexRay node performs normal frame and symbol transmission and reception
- Clock synchronization process performs offset and rate correction

The FlexRay controller enters the Normal Active state as follows:

- The completion of the startup operation (from the Startup state).

The FlexRay controller transitions from the Normal Active state as follows:

- The controller receives the CHI_HALT or CHI_FREEZE command (to the Halt state)
- The controller receives the CHI_READY command (to the Ready state)
- The controller observes internal protocol errors (to the Normal Passive state).
- The assertion of Software reset or Hardware reset (to the Reset state).

Normal Passive state

In the Normal Passive state:

- There is proper frame reception
- Clock synchronization process performs offset and rate correction
- There is no transmission of frames

The FlexRay controller enters the Normal Passive state as follows:

- The controller observes internal protocol errors (from the Normal Active state).

The FlexRay controller transitions from the Normal Passive state as follows:

- The controller receives the CHI_HALT or CHI_FREEZE command (to the Halt state).
- The controller observes internal protocol errors (to the Halt state).
- The controller receives the CHI_READY command (to the Ready state)
- The controller observes internal protocol errors (to the Normal Passive state)
- The controller observes error free communication (to the Normal Active state)
- The assertion of Software reset or Hardware reset (to the Reset state).

Halt state

The FlexRay controller enters the Halt state as follows:

- The controller observes internal protocol errors (from the Normal Passive state).
- The controller receives the CHI_HALT or CHI_FREEZE command (from the Normal Passive state, Normal Active state, Wakeup state, Startup state)

The FlexRay controller transitions from the Halt as follows:

- The controller receives the CHI_DEFAULT_CONFIG command (to the Default Config state).

Figure 2 shows the operational state transition diagram for the FlexRay controller.

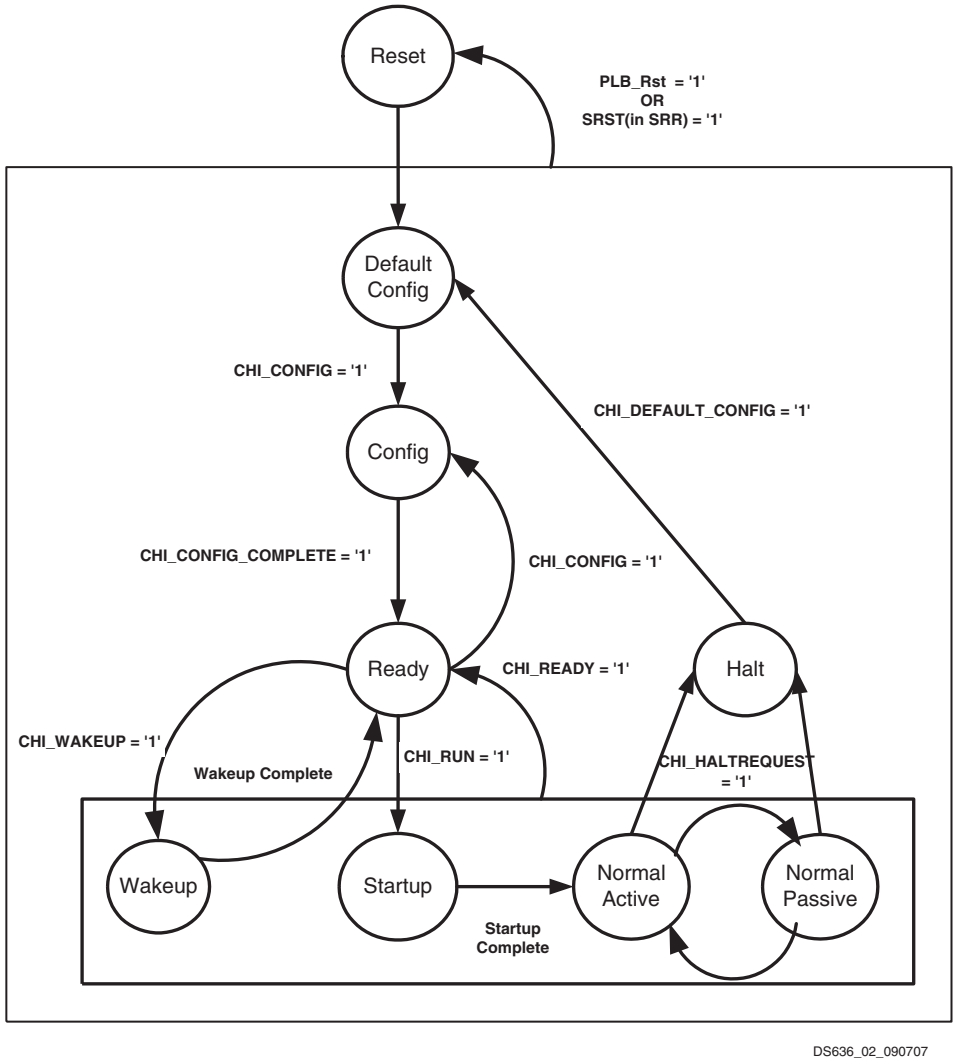

Figure 2: **Xilinx FlexRay Controller State Diagram**

I/O Signals

Table 1 defines the external I/O interface of the FlexRay controller. The external host interface of the FlexRay controller is a subset of the PLB interface. See the IBM PLB Architecture Specification version 4.6 for details on the PLB interface.

The XPS FlexRay signals are listed and described in Table 1.

Table 1: **FlexRay Controller External I/O Signals**

Port	Signal Name	Interface	I/O	Description
System Signals				
P1	SPLB_Clk	PLB	I	**PLB clock:** All host interface signals are synchronous to this clock
P2	SPLB_Rst	PLB	I	**PLB Reset**: The FlexRay core is reset to the default state upon assertion of this signal.
PLB Interface Signals				
P3	PLB_ABus[0 : 31]	PLB	I	**Address**: Bus used to specify the address being accessed either for a read or write.
P4	PLB_PAValid	PLB	I	**Select**: Indicates an active read or write access. This signal qualifies all bus inputs from the PLB master.
P5	PLB_masterID[0 : C_SPLB_MID_WIDTH - 1]	PLB	I	**Master Identification**: Indicates the identification of the master of the current transfer. Sl_MBusy, Sl_MRdErr and Sl_MWrErr must be driven using this identification.
P6	PLB_RNW	PLB	I	**Read not Write**: '1' --> Read access; '0' --> Write access.
P7	PLB_BE[0 : (C_SPLB_DWIDTH/8) - 1]	PLB	I	**Byte Enable**: Selects which byte lane of the data bus is being accessed.
P8	PLB_size[0 : 3]	PLB	I	**Transfer Size**: Indicates the size of requested tranfer. 4'h0 --> Single transfer 4'hA --> Burst transfer Others --> Not supported.
P9	PLB_type[0 : 2]	PLB	I	**Transfer Type**: Indicates the type of transfer requested. Supported transfer type is memory transfer ("000").

Table 1: **FlexRay Controller External I/O Signals**

Port	Signal Name	Interface	I/O	Description
P10	PLB_wrDBus[0 : C_SPLB_DWIDTH -1]	PLB	I	**Write Data Bus**: Data to be written to the address specified by either PLB_ABus (for single transfer) or generated address (for line, sequential and fixed length burst). The write is acknowledged by SI_wrDAck and SI_wrComp when complete.
P11	PLB_MSize[0 : 1]	PLB	I	**Master Size**: Indicates the data bus width of the associated master. “00” --> 32-bit master “01” --> 64-bit master “10” --> 128-bit master “11” --> 256-bit master. This is not supported by Xilinx
P12	PLB_wrBurst	PLB	I	PLB burst write transfer
P13	PLB_rdBurst	PLB	I	PLB burst read transfer
Unused PLB Interface Signals				
P14	PLB_UABus[0 : 31]	PLB	I	PLB upper address bits
P15	PLB_SAValid	PLB	I	PLB secondary address valid
P16	PLB_rdPrim	PLB	I	PLB secondary to primary read request indicator
P17	PLB_wrPrim	PLB	I	PLB secondary to primary write request indicator
P18	PLB_abort	PLB	I	PLB abort bus request
P19	PLB_busLock	PLB	I	PLB bus lock
P20	PLB_lockErr	PLB	I	PLB lock error
P21	PLB_wrPendReq	PLB	I	PLB pending bus write request
P22	PLB_rdPendReq	PLB	I	PLB pending bus read request
P23	PLB_wrPendPri[0 : 1]	PLB	I	PLB pending bus write request priority
P24	PLB_rdPendPri[0 : 1]	PLB	I	PLB pending bus read request priority
P25	PLB_reqPri[0 : 1]	PLB	I	PLB current request priority
P26	PLB_TAttribute[0 : 15]	PLB	I	PLB transfer attribute
PLB Slave Interface Signals				

Table 1: **FlexRay Controller External I/O Signals**

Port	Signal Name	Interface	I/O	Description
P27	Sl_addrAck	PLB	O	**Address Acknowledge**: When asserted, indicates that the address is latched. This is delayed PLB_PAValid generated as pulse. All the PLB input signals must latched as they will not be available after Sl-addrAck is asserted.
P28	Sl_SSize[0 : 1]	PLB	O	**Write Data Acknowledge**: Indicates the slave size. This is always “00” as the slave supported is 32-bit.
P29	Sl_wrDAck	PLB	O	**Write Data Acknowledge**: When asserted, indicates that the data currently on the PLB_wrDBus is no longer required
P30	Sl_wrComp	PLB	O	**Write Data Complete**: When asserted, indicates the end of the current write transfer
P31	Sl_rdDBus[0 : C_SPLB_DWIDTH - 1]	PLB	O	**Read Data**: Data to be written to the address specified by either PLB_ABus (for single transfer) or generated address (for line, sequential and fixed length burst). Data is valid when a read request followed by an acknowledge (Sl_rdDAck) is asserted. Data is mirrored for 64 and 128 data width
P32	Sl_rdDAck	PLB	O	**Read Data Acknowledge**: When asserted, indicates that the data currently on the Sl_rdDBus is valid.
P33	Sl_rdComp	PLB	O	**Read Data Complete**: When asserted, indicates that the read transfer is either complete or will be complete in the next clock cycle.

Table 1: **FlexRay Controller External I/O Signals**

Port	Signal Name	Interface	I/O	Description
P34	SI_MBusy [0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	**Busy**: Indicates that the slave is busy in performing read or write transfer. Only SI_MBusy[PLB_masterID] must be asserted and remaining bits must be driven zero
P35	SI_MWrErr [0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	**Write Error**: Indicates that the write transfer has encountered an error. Only SI_MWrErr[PLB_masterID] must be asserted and remaining bits must be driven zero.
P36	SI_MRdErr [0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	**Read Error**: Indicates that the read transfer has encountered an error. Only SI_MRdErr[PLB_masterID] must be asserted and remaining bits must be driven zero.
Unused PLB Slave Interface Signals				
P37	SI_wrBTerm	PLB	O	Slave terminate write burst transfer
P38	SI_rdWdAddr[0 : 3]	PLB	O	Slave read word address
P39	SI_rdBTerm	PLB	O	Slave terminate read burst transfer
P40	SI_MIRQ [0 : C_SPLB_NUM_MASTERS - 1]	PLB	O	Master interrupt request
P41	SI_wait	PLB	O	Slave wait
P42	SI_rearbitrate	PLB	O	Slave bus rearbitrate
Clock and Interrupt Interface Signals				
P43	Flex_Clk	FlexRay	I	FlexRay clock input (40 MHz)
P44	Sample_Clk	FlexRay	I	Sampling clock input (80MHz)
P45	Flex_Cirpt	PLB	O	FlexRay Controller Interrupt signal
P46	Flex_Txbirpt	PLB	O	FlexRay Transmit Buffer Interrupt signal
P47	Flex_Rxbirpt	PLB	O	FlexRay Receive Buffer Interrupt signal

Table 1: **FlexRay Controller External I/O Signals**

Port	Signal Name	Interface	I/O	Description
P48	Flex_Rxfirpt	PLB	O	FlexRay Receive FIFO Interrupt signal
FlexRay PHY Interface Signals				
P49	TXD	FlexRay	O	**Transmit PHY Line:** The Transmit PHY line connected to the Flexray PHY chip
P50	TXD_EN	FlexRay	O	**Transmit Enable PHY Line:** The Transmit Enable signal connected to the FlexRay PHY chip. When '0' indicates an active transmission on the TxD line.
P51	RXD	FlexRay	I	**Receive PHY Line:** The Receive PHY line connected to the FlexRay controller. The source of RxD is the FlexRay PHY chip.

Design Parameters

To obtain a Xilinx FlexRay controller that is uniquely tailored to meet minimum system requirements, certain features are parameterized. This results in a design using only the required resources, there by providing the best possible performance. Table 2 shows Xilinx FlexRay controller features that can be parameterized.

Table 2: **Xilinx FlexRay Controller Design Parameters**

Parameter Name	Parameter Description	Allowable Values	Default Value
C_BASEADDR	Base Address of the Xilinx FlexRay Controller	'XXXXXXXXXXXXXXXX0000000000000000' where x can be user controlled	0X80010000
C_HIGHADDR	High Address of the Xilinx FlexRay Controller	'XXXXXXXXXXXXXXXX0111111111111111' where x is identical to the value in C_BASEADDR	0X80017FFF
C_FAMILY	Target FPGA family	Spartan-3, Spartan-3XA, Spartan-3A, Spartan-3A DSP, Spartan-3E, Spartan-3EXA, Virtex-II PRO, Virtex-4	spartan3

Table 2: **Xilinx FlexRay Controller Design Parameters**

Parameter Name	Parameter Description	Allowable Values	Default Value
C_CHANNEL	Channel Selection A or B	true/false true - channel A false - channel B	true
C_MAX_PAYLOAD	Maximum Payload Size (bytes)	4,8,12,16.....252,256	4
C_NUM_TX_BUF	Number of Transmit buffers	2,4,8,16,32,64,128	2
C_NUM_RX_BUF	Number of Receive buffers	2,4,8,16,32,64,128	2
C_RX_FIFO_DPTH	Depth of Receive FIFO	2,4,8,16,32,64,128	2
C_SPLB_MID_WIDTH	PLB master ID width	$\log_2$(C_SPLB_NUM_MASTERS)	1
C_SPLB_NUM_MASTERS	Number of PLB masters	1-16	1
C_SPLB_DWIDTH	Data width for XPS FlexRay and PLB	32,64,128	32

Base Address

The user can configure the Base Address of the core to `'XXXXXXXXXXXXXXXX0000000000000000'` where x can take a '0' or a '1'.

High Address

The user can configure the High Address of the core to `C_BASEADDR` + `0x00007FFF`.

Channel Selection

The user can configure the single channel FlexRay controller to support either FlexRay Channel A or FlexRayChannel B.

Maximum Payload Size

The user can configure the payload size from 0 bytes to 256 bytes in multiples of 4 bytes. The 32-bit PLB bus interface supports only word accesses, therefore even though the maximum payload size allowed by the Protocol Specification V2.1 Rev A is 254 bytes, the physical storage allocated to store the maximum payload is fixed to 256 bytes. The value chosen for `Max Payload Size` must be greater than or equal to the protocol parameters `gPayloadLengthStatic` and `pPayloadLengthDynMax`.

Number of Transmit buffers

The number of the Tx Buffers can be configured from 2 to 128 in incremental powers of 2. Each Tx Buffer consists of 3 dedicated header registers. A single BRAM is used to store the header registers for all the Tx Buffers.

The payload associated with each Tx Buffer is stored in a Payload Buffer. Depending on the number of Tx Buffers and the maximum payload size chosen, the size of the Payload Buffer can range from 2 KBytes to 8 KBytes with an incremental step size of 2 KBytes.

Number of Receive buffers

The number of the Rx Buffers can be configured from 2 to 128 in incremental powers of 2. The maximum memory allocated to store Rx Buffers is 8 KBytes.

The upper limit of 8 KBytes for the storage of Rx Buffers places certain restrictions on the number of Rx Buffers and the maximum payload size that can be configured.The product of `Max Payload Size` and `Number of Rx Buffers` should be less than 8KBytes.

Depth of the Receive FIFO

The depth of the Rx FIFO can be configured from 1 to 128 in incremental powers of 2. The maximum memory allocated for the Rx FIFO is 8 KBytes.

The upper limit of 8KBytes on the size of the Rx FIFO places certain restrictions on the depth of Rx FIFO and the maximum payload size that can be configured. The product of "`Max Payload Size` + 8" and `Rx FIFO Depth` should be less than 8KBytes.

PLB Master ID Width

PLB Master ID Bus Width is equal to $\log_2$ of the number of PLB masters connected to the PLB Bus or 1, whichever is greater.

Number of PLB Masters

This parameter is equal to the number of masters connected to the PLB Bus.

Data Width for PLB

This integer parameter is used by the PLB slave to size PLB data bus related components within the Slave attachment. This value must be set to match the actual width of the PLB bus, 32, 64 or 128-bits.

Register Descriptions

Table 3 lists Xilinx FlexRay controller registers. Each of these registers is 32 bits wide and is represented in big-endian format. Bit '0' represents MSB and bit '31' represents LSB. Read operations to address locations marked as Reserved in the memory map, and reserved bits and bit fields in valid address locations return garbage. Writes to address locations marked as Reserved in the memory map are prohibited. For valid address locations, '0's should be written to reserved bits and bit fields.

The FlexRay controller registers are grouped based on the sub-modules in which they are used. See *Functional Description* section of this document for a brief description of the sub-modules in the Xilinx FlexRay controller.

Values for the protocol specific registers and the formulas to compute them can be obtained from Appendix B in the *FlexRay Protocol Specification v2.1 Rev A*.

Table 3: **Xilinx FlexRay Controller Registers**

Register Name	Address	Access
Protocol Engine Registers		
Software Reset Register (SRR)	0x00000000	Read/Write
Static Slot Duration Register (SSDR)	0x00000004	Read/Write

Register Name	Address	Access
Number of Static Slots Register (NSSR)	0x00000008	Read/Write
Static Segment Payload Length Register (SSPLR)	0x0000000C	Read/Write
Action Point Offset Register (APOR)	0x00000010	Read/Write
Startup Frame Configuration Register (STFCR)	0x00000014	Read/Write
Sync Frame Configuration Register (SYFCR)	0x00000018	Read/Write
Network Management Vector Length Register (NMVLR)	0x0000001C	Read/Write
Reserved	**0x00000020 - 0x0000003C**	Read/Write
POC Module Registers		
CHI Command Register (CR)	0x00000040	Read
CHI Status Register (CSR)	0x00000044	Read
Allow Halt Dueto Clock Register (AHCR)	0x00000048	Read/Write
Slot Mode Register (SMR)	0x0000004C	Read/Write
Maximum without Clock Correction Fatal Register(MCCFR)	0x00000050	Read/Write
Maximum without Clock Correction Passive Register(MCCPR)	0x00000054	Read/Write
PAllow Passive to Active Register (PAPAR)	0x00000058	Read/Write
VAllow Passive to Active Status Register (VAPASR)	0x0000005C	Read
Clock Correction Failed Status Register (CCFSR)	0x00000060	Read
Reserved	**0x00000064 - 0x00000078**	Read/Write
Test Mode Register (TMR)	0x0000007C	Write
Reserved	**0x00000080 - 0x000000FC**	Read/Write
CODEC Module Registers		
TSS Register (TSSR)	0x00000100	Read/Write
WUP Receive Idle Register (WRIR)	0x00000104	Read/Write
WUP Receive Low Register (WRLR)	0x00000108	Read/Write
WUP Window Register(WRWR)	0x0000010C	Read/Write
WUP Transmit Idle Register (WTIR)	0x00000110	Read/Write
WUP Transmit Low Register (WTLR)	0x00000114	Read/Write
CAS Receive Max Register (CRMXR)	0x00000118	Read/Write
Decoding Correction Register (DCR)	0x0000011C	Read/Write

Register Name	Address	Access
Wakeup Pattern Sequence Register (WPSR)	0x00000120	Read/Write
Reserved	**0x00000124 - 0x0000013C**	Read/Write
MAC Module Registers		
Sync Slot Register (SYSR)	0x00000140	Read/Write
Minislot Duration Register (MSDR)	0x00000144	Read/Write
Minislot Action Point Offset Register (MAPR)	0x00000148	Read/Write
Dynamic Slot Idle Phase Register (DSIPR)	0x0000014C	Read/Write
Number of Minislots Register (NMSR)	0x00000150	Read/Write
Latest Tx Register (LTR)	0x00000154	Read/Write
Symbol Window Duration Register (SWDR)	0x00000158	Read/Write
MTS Transmit Register (MTR)	0x0000015C	Read/Write
Slot Counter Status Register (SCSR)	0x00000160	Read
Last Dynamic Frame Transmit Status Register (LDFTSR)	0x00000164	Read
Reserved	**0x00000168 - 0x000001FC**	Read/Write
Clock Sync Module Registers		
Extern Offset Correction Register (EOCR)	0x00000200	Read/Write
Extern Rate Correction Register (ERCR)	0x00000204	Read/Write
Apply Extern Correction Register(AECR)	0x00000208	Read/Write
Offset Correction Limit Register (OCLR)	0x0000020C	Read/Write
Rate Correction Limit Register (RCLR)	0x00000210	Read/Write
Number Of Microticks Per Cycle Register (NMICR)	0x00000214	Read/Write
Number Of Macroticks Per Cycle Register (NMACR)	0x00000218	Read/Write
Maximum Drift Register (MDR)	0x0000021C	Read/Write
Cluster Drift Damping Register (CDDR)	0x00000220	Read/Write
Macro Initial Offset Register (MAOR)	0x00000224	Read/Write
Offset Correction Start Register (OCSR)	0x00000228	Read/Write
Micro Initial Offset Register (MIOR)	0x0000022C	Read/Write
Sync Node Max Register (SNMR)	0x00000230	Read/Write

Register Name	Address	Access
Accepted Startup Range Register (ASRR)	0x00000234	Read/Write
Reserved	**0x00000238 - 0x0000023C**	Read/Write
Rate Correction Status Register (RCSR)	0x00000240	Read
Offset Correction Status Register (OFCSR)	0x00000244	Read
Valid Sync Frames Status Register (VSFSR)	0x00000248	Read
Cycle Count Status Register (CCSR)	0x0000024C	Read
Macrotick Status Register (MSR)	0x00000250	Read
Reserved	**0x00000254 - 0x000002FC**	Read/Write
Sync Frames Status Register[1..30] (SFSR[n])	0x00000300-0x00000374	Read
Reserved	**0x00000378 - 0x000003FC**	Read/Write
WakeUp Module Registers		
Listen Timeout Register (LTOR)	0x00000400	Read/Write
Listen Noise Register (LNR)	0x00000404	Read/Write
Number of ColdStart Attempts Register (NCAR)	0x00000408	Read/Write
Wakeup Status Register (WSR)	0x0000040C	Read
ColdStart Noise Status Register (CNSR)	0x00000410	Read
Remaining Coldstart Attempts Status Register (RCASR)	0x00000414	Read
Startup Status Register (STSR)	0x00000418	Read
Reserved	**0x0000041C - 0x0000043C**	Read/Write
Timer Registers		
Absolute Timer Control Register(ATCR)	0x00000440	Read/Write
Absolute Timer Cycle Count Register(ATCCR)	0x00000444	Read/Write
Absolute Timer Macrotick Offset Register(ATMOR)	0x00000448	Read/Write
Relative Timer Control Register(RTCR)	0x0000044C	Read/Write
Relative Timer Macrotick Offset Register (RTMOR)	0x00000450	Read/Write
Reserved	**0x00000454 - 0x0000045C**	Read/Write
Controller Status Registers		
Aggregated Status Register (ASR)	0x00000460	Read
NIT Status Register (NSR)	0x00000464	Read

Register Name	Address	Access
Symbol Window Status Register (SWSR)	0x00000468	Read
Reserved	**0x0000046C**	Read/Write
Network Management Vector Registers[1..3] (NMVR[n])	0x00000470 - 0x00000478	Read
Reserved	**0x0000047C - 0x000004FC**	Read/Write
Receive Slot Status Registers[1..512] (RSSR[n])	0x00000500-0x00000CFC	Read
Reserved	**0x00000D00 - 0x00000EFC**	Read/Write
Interrupt Control and Status Registers		
Controller Interrupt Status Register (CISR)	0x000000F00	Read
Controller Interrupt Enable Register(CIER)	0x000000F04	Read/Write
Controller Interrupt Clear Register (CICR)	0x000000F08	Write
Reserved	**0x00000F0C**	Read/Write
Transmit Buffer Interrupt Status Register[1..4] (TXBISR[n])	0x000000F10 - 0x000000F1C	Read
Transmit Buffer Interrupt Enable Register[1..4] (TXBIEN[n])	0x000000F20 - 0x000000F2C	Read/Write
Transmit Buffer Interrupt Clear Register[1..4] (TXBICR[n])	0x000000F30 - 0x000000F3C	Write
Receive Buffer Interrupt Status Register[1..8] (RXBISR[n])	0x000000F40 - 0x000000F5C	Read
Receive Buffer Interrupt Enable Register[1..8] (RXBIER[n])	0x000000F60 - 0x000000F7C	Read/Write
Receive Buffer Interrupt Clear Register[1..8] (RXBICR[n])	0x000000F80 - 0x000000F9C	Write
Receive FIFO Interrupt Status Register (RXFISR)	0x000000FA0	Read
Receive FIFO Interrupt Enable Register (RXFIER)	0x000000FA4	Read/Write
Receive FIFO Interrupt Clear Register (RXFICR)	0x000000FA8	Write
Reserved	**0x00000FAC - 0x00000FFC**	**Read/Write**
Transmit buffers		
Transmit buffer Payload Storage	0x00001000-0x00002FFC	Read/Write
Transmit buffer Header 1 Registers [1..128] (TXBH1R[n])	0x00003000-0x00003FFC	Read/Write

Register Name	Address	Access
Transmit buffer Header 2 Registers [1..128] (TXBH2R[n])	0x00003000-0x00003FFC	Read/Write
Transmit buffer Header 3 Registers [1..128] (TXBH3R[n])	0x00003000-0x00003FFC	Read/Write
Receive Buffers		
Receive Buffer[1..66] Registers[1..128] (RXB[y]R[n])	0x00004000-0x00005FFC	Read
Receive FIFO		
Receive FIFO [1..66] Registers (RXF[y]R)	0x00006000-0x00006108	Read
Reserved	**0x0000610C - 0x000061FC**	**Read/Write**
Receive FIFO Threshold Register (RFTR)	0x00006200	Read/Write
Reserved	**0x00006204 - 0x000062FC**	**Read/Write**
Acceptance Filters		
Receive Buffer Acceptance Filter 1 Registers [1..128] (RBAF1R[n])	0x00006300-0x000066FC	Read/Write
Receive Buffer Acceptance Filter 2 Registers [1..128] (RBAF2R[n])	0x00006300-0x000066FC	Read/Write
Receive FIFO Acceptance Filter ID 1 Register [1..4] (RFAFI1R[n])	0x00006700-0x0000671C	Read/Write
Receive FIFO Acceptance Filter Mask 1 Register [1..4] (RFAFM1R[n])	0x00006700-0x0000671C	Read/Write
Receive FIFO Acceptance Filter ID 2 Register [1..4] (RFAFI2R[n])	0x00006720-0x0000673C	Read/Write
Receive FIFO Acceptance Filter Mask 2 Register [1..4] (RFAFM2R[n])	0x00006720-0x0000673C	Read/Write

Protocol Engine Registers

Software Reset Register (SRR)

The SRST bit in the SRR provides the following functionality:

- **Acts as a Software Reset**. Writing a '1' to the SRST bit places the FlexRay controller in the reset state.
- **Acts as a status indicator**. It is set to '1' when the FlexRay controller is in the reset state. The FlexRay controller will enter the reset state when the system reset (`PLB_Rst signal`) is asserted.

Operation of Software Reset

The Software Reset resets the CHI and PE modules. Writing a '1' to the `SRST` bit places the FlexRay controller in a reset state. Upon completion of the reset operation, the FlexRay controller transitions to the default Config state. During the time when the FlexRay controller is in the Reset state, reads to the `SRST` bit return a '1.' When the FlexRay controller transitions to the Default Config state, reads to the `SRST` bit return a '0.' The application of Software Reset during normal operation of the FlexRay

controller immediately places the node into Reset state and current frame transmission or reception is immediately stopped.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Limitations of Software Reset

The Software Reset does not clear the contents of the block RAMs used in the design.

Table 4 shows the bit positions in the SRR and Table 5 provides the SRR description

Table 4: **Software Reset Register Bit Positions**

0-30	31
RESERVED	SRST

Table 5: **Software Reset Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	SRST	Read/Write	0	**Software Reset**: The Software Reset bit for the FlexRay controller. Writes: '1' = Writing a '1' to this bit places the FlexRay controller in the reset state. Reads: '1' = Indicates that the FlexRay controller is in the reset state. The FlexRay controller can enter the Reset state when the Software Reset is asserted or when the System Reset is asserted. '0' = Indicates that the FlexRay controller is not in the reset state.

Static Slot Duration Register (SSDR)

The Static Slot Duration Register (SSDR) corresponds to the protocol parameter `gdStaticSlot`. This is a global cluster parameter and indicates the number of macroticks in a static slot. It ranges from 4–661 macroticks.

Access Criterion for Reads/Writes

- Reads: Allowed in all operational states.

- Writes: Allowed only in Config state.

Table 6 shows the bit positions in the SSDR and Table 7 provides the SSDR description

Table 6: **Static Slot Duration Register Bit Positions**

0-20	21-31
RESERVED	GDSTATICSLOT(9:0)

Table 7: **Static Slot Duration Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
22-31	GDSTATICSLOT(9:0)	Read/Write	128 Macroticks	**gdStaticSlot**: Indicates the number of macroticks that constitute the duration of a static slot.

Number Of Static Slots Register (NSSR)

Every FlexRay communication cycle must contain a static segment. The Number of Static Slots Register (NSSR) corresponds to the `gNumberOfStaticSlots` protocol parameter. This is a global cluster parameter and indicates the number of static slots in the static segment. It ranges from two to `cStaticSlotIDMax` (1023).

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 8 shows the bit positions in the NSSR and Table 9 provides the NSSR description.

Table 8: **Number Of Static Slots Register Bit Positions**

0-21	22-31
RESERVED	GNUMSTATICSLOTS(9:0)

Table 9: **Number Of Static Slots Register Description**

Bits	Name	Access	Default Value	Description
0-21	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
22-31	GNUMSTATICSLOTS(9:0)	Read/Write	8	**gNumberOfStaticSlots**: Indicates the number of static slots in the static segment.

Static Segment Payload Length Register (SSPLR)

In the static segment, all the frames have the same payload length— equal to `gPayloadLengthStatic`. The Static Segment Payload Length Register (SSPLR) corresponds to the `gPayloadLengthStatic` protocol parameter. This parameter is a global cluster parameter and indicates the payload length (in multiples of two bytes) for static frames. It ranges from zero to `cPayloadLengthMax` (127 two-byte words).

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 10 shows the bit positions in the SSPLR and Table 11 provides the SSPLR description.

Table 10: **Static Segment Payload Length Register Bit Positions**

0-24	25-31
RESERVED	GPLSTATIC(6:0)

Table 11: **Static Segment Payload Length Register Description**

Bits	Name	Access	Default Value	Description
0-24	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
25-31	GPLSTATIC(6:0)	Read/Write	16	**gPayloadLengthStatic**: Indicates the payload length in multiples of 2 bytes.

Action Point Offset Register (APOR)

Each static slot contains an action point that is offset from the start of the slot by `gdActionPointOffset` macroticks. In the static segment, frame transmissions start at the action point of the static slot. During Symbol window, symbol transmissions start at the action point of the symbol window.

The Action Point Offset Register (APOR) corresponds to the `gdActionPointoffset` protocol parameter. This is a global cluster parameter and indicates the action point offset in macroticks. The APOR can take values from 1 to 63 macroticks.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 12 shows the bit positions in the APOR and Table 13 provides the APOR description

Table 12: **Action Point Offset Register Bit Positions**

0-25	26-31
RESERVED	GAPOFFSET(5:0)

Table 13: **Action Point Offset Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved:** These bit positions are reserved for future expansion.
26-31	GAPOFFSET(5:0)	Read/Write	3	**gdActionPointOffset:** Indicates the number of macroticks the action point is offset from the beginning of a static slot or symbol window.

Startup Frame Configuration Register (STFCR)

The Startup Frame Configuration Register (STFCR) corresponds to the `pKeySlotUsedForStartup` node parameter. This is a node-specific parameter and indicates whether the node can transmit a startup frame.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 14 shows the bit position in STFCR and Table 15 provides the STFCR description.

Table 14: **Startup Frame Configuration Register Bit Positions**

0-30	31
RESERVED	PKSUP

Table 15: **Startup Frame Configuration Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	PKSUP	Read/Write	0	**pKeySlotUsedForStartup**: This bit determines whether a startup frame is transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. '0' - A startup frame is not transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. '1' - A startup frame is transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. When the PKSUP bit is set to '1,' the PKSYNC bit in the SYFCR must be set to '1.'

Sync Frame Configuration Register (SYFCR)

The Sync Frame Configuration Register (SYFCR) corresponds to the `pKeySlotUsedForSync` node parameter. This is a node-specific parameter and indicates whether the node can transmit a Sync Frame.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 16 shows the bit positions in the SYFCR and Table 17 provides the SYFCR description

Table 16: **Sync Frame Configuration Register Bit Positions**

0-30	31
RESERVED	PKSYNC

Table 17: **Sync Frame Configuration Register Bit Field Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	PKSYNC	Read/Write	0	**pKeySlotUsedForSync**: This bit determines whether a sync frame is transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. '0' - A sync frame is not transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. '1' - A sync frame is transmitted during the slot indicated by the PKEYSLOTID field in the SYSR. When the PKSUP bit in the STFCR is set to '1,' the PKSYNC bit must be set to '1.'

Network Management Vector Length Register (NMVLR)

The Network Management Vector Length Register (NMVLR) corresponds to the `gNetworkManagementVectorLength` node parameter. This is a global cluster parameter that indicates the length of the Network Management Vector. The NMVLR can accept values from 0 to 12 bytes.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 18 shows the bit positions in the NMVLR and Table 19 provides the NMVLR description

Table 18: **Network Management Vector Length Register Bit Positions**

0-27	28-31
RESERVED	NMVL[3..0]

Table 19: **Network Management Vector Length Register Bit Field Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	NMVL[3..0]	Read/Write	0	**gNetworkManagementVectorLength**: Indicates the length of the Network Management Vector.

POC Module Registers

CHI Command Register (CR)

The CHI Command Register (CR) contains the Host Commands as specified in the *FlexRay Protocol Specification Rev 2.1 A*.

The Command Register defines the following Host Commands:

1. CHI_ALL_SLOTS : The CHI_ALL_SLOTS command causes the node to transition from the Single Slot transmission mode to the All Slot transmission mode at the end of the current communication cycle.
2. CHI_ALLOW_COLDSTART: The CHI_ALLOW_COLDSTART command causes the node to initiate the cold start mechanism immediately.
3. CHI_CONFIG : The CHI_CONFIG command causes the operational state of the node to change to Config state from either Ready state or Default Config state immediately.
4. CHI_CONFIG_COMPLETE: The CHI_CONFIG_COMPLETE command causes the operational state of the node to change to Ready state from Config state immediately.
5. CHI_DEFAULT_CONFIG: The CHI_DEFAULT_CONFIG command causes the operational state of the node to change to Defualt Config state from Halt state immediately.
6. CHI_FREEZE : The CHI_FREEZE command causes the operational state of the node to change to Halt State immediately.
7. CHI_HALT: The CHI_HALT command causes the operational state of the node to change to Halt state from Normal Active state or Normal Passive state at the end of the current communication cycle.
8. CHI_READY: The CHI_READY command causes the operational state of the node to change to Ready state from Wakeup state, Startup state, Normal Active state or Normal Passive state immediately.
9. CHI_RUN: The CHI_RUN command causes the operational state of the node to change to Startup state from Ready state immediately.
10. CHI_WAKEUP: The CHI_WAKEUP command causes the operational state of the node to change to Wakeup state from Ready state immediately.

Access Criterion for Reads/Writes

- Reads—Not allowed to this register.
- Writes—Allowed in all operational states. Writes are allowed only when the CHIBSY bit in the CSR is '0.'

In addition the following restrictions apply:

- The CHI_ALL_SLOTS command must be issued only during the Normal Active state or Normal Passive state.
- The CHI_ALLOW_COLDSTART command must not be issued during the following states: Default Config state, Config state, Halt state.
- The CHI_CONFIG command must be issued only during the Default Config state or Ready state.
- The CHI_CONFIG_COMPLETE command must be issued only during the Config state.
- The CHI_DEFAULT_CONFIG command must be issued only during the Halt state.
- The CHI_FREEZE command can be issued during any operational mode.
- The CHI_HALT command must be issued only during the Normal Active state or Normal Passive state.
- The CHI_READY command must not be issued during the following states: Default Config state, Config state, Ready state, Halt state.
- The CHI_RUN command must be issued only during the Ready state.
- The CHI_WAKEUP command must be issued only during the Ready state.

The FlexRay controller does not provide any write protection scheme for write access to the CR. The user application must ensure that the access criterion for the CR are followed.

Table 20 shows the bit positions in the CR and Table 21 provides the CR description

Table 20: **CHI Command Register Bit Positions**

0-27	28-31
RESERVED	CHICMD(3:0)

Table 21: **CHI Command Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	CHICMD(3:0)	Write only	0	**CHI Command**: The Host commands are as follows: '0000' - CHI_ALL_SLOTS '0001' - CHI_ALLOW_COLDSTART '0010" - CHI_CONFIG '0011' - CHI_CONFIG_COMPLETE '0100' - CHI_DEFAULT_CONFIG '0101' - CHI_FREEZE '0110' - CHI_HALT '0111' - CHI_READY '1000' - CHI_RUN '1001' - CHI_WAKEUP

CHI Status Register (CSR)

The CHI Status Register (CSR) contains POC status information as specified in the *FlexRay Protocol Specification Rev 2.1 A*. The CSR defines the following status fields:

- vPOC!State
- vPOC!Freeze
- vPOC!CHIHaltRequest
- vPOC!ColdstartNoise
- vPOC!SlotMode
- vPOC!ErrorMode

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 22 shows the bit positions in the CSR and Table 23 provides the CSR description

Table 22: **CHI Status Register Bit Positions**

0-21	22-23	24-25	26	27	28-30	31
RESERVED	POCEM(1:0)	POCSM(1:0)	POCHRQ	POCFRZ	POCSTATE(2:0)	CHIBSY

Table 23: **CHI Status Register Description**

Bits	Name	Access	Default Value	Description
0-21	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
22-23	POCEM(1:0)	Read	"01"	**vPOC!ErrorMode**: Indicates the current error mode. **POCEM** is enumerated as follows: '00' = **Reserved** '01' = **ACTIVE**. Indicates the current error mode is active. '10' = **PASSIVE**. Indicates that the current error mode is passive. '11' = **COMM_HALT**. Indicates that the current error mode is halt. **POCEM** is '01' when the node is in Default Config state.
24-25	POCSM(1:0)	Read	"01"	**vPOC!SlotMode**: Indicates the current slot mode. POCSM is enumerated as follows: '00' = **Reserved** '01' = **SINGLE**. Indicates that the current slot mode is Single Slot '10' = **ALL_PENDING**. Indicates that the transition to All Slot from Single Slot is pended until the end of the current communication cycle. This transition is initiated using the CHI_ALL_SLOTS command. '11' = **ALL**. Indicates that the current slot mode is All Slot. **POCSM** is '01' when the node is in Default Config state.

Bits	Name	Access	Default Value	Description
26	POCHRQ	Read	0	**vPOC!HaltRequest**: Indicates a pended CHI_HALT command. '0' - Indicates that the CHI_HALT command is not pended for execution. '1' - Indicates that the CHI_HALT command has been issued and the command is pended for execution at the end of the current communication cycle. This bit is cleared when the node is in Default Config state.

Bits	Name	Access	Default Value	Description
27	POCFRZ	Read	0	**vPOC!Freeze**: Indicates that node has entered Halt state. '0' - Indicates that the operational state of the node is not Halt state. '1' - Indicates that the operational state of the node is Halt state. This bit is cleared when the node is in Default Config state.
28-30	POCSTATE(2:0)	Read	"001"	**vPOC!State**: Indicates the current operational state of the FlexRay controller. CHISTATUS is enumerated as follows: '000' = Indicates Config state. '001' = Indicates Default Config state/ Reset state. '010' = Indicates Halt state. '011' = Indicates Normal Active state. '100' = Indicates Normal Passive state. '101' = Indicates Ready state. '110' = Indicates Wakeup state. '111' = Indicates Startup state. When the CHI_FREEZE command is asserted, the CHISTATUS reflects the current operational state of the controller. In this case, the CHISTATUS remains unchanged even after the transition to the Halt state. When the CHI_DEFAULT_CONFIG command is asserted in the Halt state, the CHISTATUS is updated to reflect the Default Config state of operation.
31	CHIBSY	Read	0	**CHI Command Register BUSY**: When set to '1,' indicates that the CR cannot be written to. This bit is set to a '1' after the CR is written to. This bit is automatically cleared by the FlexRay controller. '0' = The CHI Command Register can be written to. '1' = The CHI Command Register cannot be written to.

Allow Halt Due To Clock Register (AHCR)

The Allow Halt Due to Clock Register (AHCR) corresponds to the `pAllowHaltDueToClock` protocol parameter. This is a node-specific parameter that governs transitions to Halt state due to clock synchronization errors.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 24 shows the bit positions in the AHCR and Table 25 provides the AHCR description.

Table 24: **Allow Halt Due To Clock Register Bit Positions**

0-30	31
RESERVED	PAHDTC

Table 25: **Allow Halt Due To Clock Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
31	PAHDTC	Read/Write	0	**pAllowHaltDueToClock**: Controls the transition to the Halt state due to a clock synchronization errors. '0' = The controller does not transition to Halt state due to clock synchronization errors. '1' = The controller will transition to Halt state due to clock synchronization errors.

Slot Mode Register (SMR)

The FlexRay controller can operate in either Single Slot or All Slot modes. In Single Slot mode, the controller transmits a single frame per communication cycle in the slot indicated by the PKEYSLOTID field in the SYSR. In the All Slot mode, frames are transmitted in all the transmission slots assigned to the node.

In the Single Slot mode, the following frame transmissions can occur:

- A Startup Frame is transmitted in the slot indicated by the PKEYSLOTID field in the SYSR—if the PKSUP bit in the STFCR and the PKSYNC bit in the SYFCR are a '1.'
- A Sync Frame is transmitted in the slot indicated by the PKEYSLOTID field in the SYSR—if the PKSYNC bit in the SYFCR is '1' and the PKSUP bit in the STFCR is '0.'
- A normal frame is transmitted in the slot indicated by the PKEYSLOTID field in the SYSR—if both the PKSUP bit in the STFCR and the PKSYNC bit in the SYFCR are '0.'

If the FlexRay controller is in Single Slot mode (the PSSE bit is a '0'), issuing a `CHI_ALL_SLOTS` command causes the transition to All Slot mode at the end of the current communication cycle.

However, the PSSE bit is not updated by the controller. Instead, the current slot mode of the controller is indicated by the POCEM field in the CSR.

The Slot Mode Register (SMR) corresponds to the `pSingleSlotEnabled` protocol parameter. This is a node-specific parameter that determines whether the node can transmit in only one slot or in all configured slots after Startup state.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 26 shows the bit positions in the SMR and Table 27 provides the SMR description.

Table 26: **Slot Mode Register Bit Positions**

0-30	31
RESERVED	PSSE

Table 27: **Slot Mode Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	PSSE	Read	0	**pSingleSlotEnabled**: Indicates the Single Slot transmission mode or the All Slot transmission mode. '1' = Indicates the Single Slot transmission mode. '0' = Indicates the All Slot transmission mode.

Maximum Without Clock Correction Fatal Register (MCCFR)

The Maximum Without Clock Correction Fatal Register (MCCFR) corresponds to the `gMaxWithoutClockCorrectionFatal` protocol parameter. This is a global cluster parameter that defines the maximum allowable number of consecutive even and odd cycle pairs with missing clock correction terms. When that number is reached, the operational state of the controller transitions from Normal Active state or Normal Passive state to a Halt state.

The value written into the MCCFR should be greater than or equal to the value written into the MCCPR. The maximum value that the MCCFR can take is 15.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 28 shows the bit positions in the MCCFR and Table 29 provides the MCCFR description.

Table 28: **Maximum Without Clock Correction Fatal Register Bit Positions**

0-27	28-31
RESERVED	GMWCCF(3:0)

Table 29: **Maximum Without Clock Correction Fatal Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	GMWCCF(3:0)	Read/Write	15	**gMaxWithoutClockCorrectionFatal:** Indicates the maximum number of consecutive even/odd cycle pairs with missing clock correction terms that can be tolerated before transition to the Halt state of operation.

Maximum Without Clock Correction Passive Register (MCCPR)

The Maximum Without Clock Correction Passive Register (MCCPR) corresponds to the `gMaxWithoutClockCorrectionPassive` protocol parameter. This is a global cluster parameter that defines the maximum number of consecutive even and odd cycle pairs with missing clock correction terms that can occur before the controller transitions from Normal Active to Normal Passive state.

The value written into the MCCPR should be less than or equal to the value written into the MCCFR. The maximum number that the MCCPR can hold is 15.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 30 shows the bit positions in the MCCPR and Table 31 provides the MCCPR description.

Table 30: **Maximum Without Clock Correction Passive Register Bit Positions**

0-27	28-31
RESERVED	GMWCCP(3:0)

Table 31: **Maximum Without Clock Correction Passive Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
28-31	GMWCCP(3:0)	Read/Write	15	**gMaxWithoutClockCorrectionPassive:** Indicates the maximum number of consecutive even/odd cycle pairs with missing clock correction terms that can be tolerated before transition to Normal Passive state of operation.

PAllow Passive To Active Register (PAPAR)

The PAllow Passive to Active Register (PAPAR) corresponds to the `pAllowPassiveToActive` protocol parameter. This is a node-specific parameter that defines the maximum number of consecutive even and odd cycle pairs with valid clock correction terms that can occur before the controller transitions from Normal Passive state to Normal Active state.

The PAPAR accepts values from 0 to 31. A zero value indicates that the controller will not transition from Normal Passive to the Normal Active state.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 32 shows the bit positions in the PAPAR and Table 33 provides the PAPAR description.

Table 32: **PAllow Passive To Active Register Bit Positions**

0-26	27-31
RESERVED	PAPTA(4:0)

Table 33: **PAllow Passive To Active Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
27-31	PAPTA(4:0)	Read/Write	0	**pAllowPassiveToActive**: Indicates the maximum number of consecutive even/odd cycle pairs with valid clock correction terms that need to seen before transition to the Normal Active state of operation from the Normal Passive state of operation.

VAllow Passive to Active Status Register (VAPASR)

The VAllow Passive to Active Status Register (VAPASR) is a status register that corresponds to the `vAllowPassiveToActive` protocol status parameter. It defines the number of consecutive even and odd cycle pairs with valid clock correction terms observed when the node is in Normal Passive state.

The VAPASR accepts values from 0 to 31.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 34 shows the bit positions in the VAPASR and Table 35 provides the VAPASR description.

Table 34: **VAllow Passive To Active Status Register Bit Positions**

0-26	27-31
RESERVED	VAPTA(4:0)

Table 35: **VAllow Passive To Active Status Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
27-31	VAPTA(4:0)	Read	0	**vAllowPassiveToActive**: Represents the status counter that indicates the number of consecutive even/odd cycle pairs with valid clock correction terms that are observed in the Normal Passive state of operation.

Clock Correction Failed Status Register (CCFSR)

The Clock Correction Failed Status Register (CCFSR) is a status register that corresponds to the `vClockCorrectionFailed` protocol status parameter. It indicates the number of consecutive even and odd cycle pairs where rate and offset corrections were not performed.

The CCFSR accepts values from 0 to 15.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 36 shows the bit positions in the CCFSR and Table 37 provides the CCFSR description.

Table 36: **Clock Correction Failed Status Register Bit Positions**

0-27	28-31
RESERVED	VCCF(3:0)

Table 37: **Clock Correction Failed Status Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	VCCF(3:0)	Read	0	**vClockCorrectionFailed**: Indicates the number of consecutive even/odd cycle pairs where rate and offset corrections were not performed.

Test Mode Register (TMR)

The Test Mode Register (TMR) contains a single bit called BYPWUS. The BYPWUS bit is can be used only for functional and timing simulations to by pass the normal wakeup and startup procedures.

Access Criterion for Reads/Writes

- Reads—Not allowed.
- Writes— Allowed only when the core is being used in a functional/timing simulation environment.

Table 38 shows the bit positions in the CCFSR and Table 39 provides the CCFSR description.

Table 38: **Test Mode Register Bit Positions .**

0-30	31
RESERVED	BYPWUS

Table 39: **Test Mode Register Bit Description**

Bit	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	BYPWUS	Write	0	**BYPWUS**: Bypass Wakeup and Startup. '1' - The Wakeup and Startup procedures are bypassed and the node enters normal active mode of operation. '0' - The Bypass mechanism is disabled.

CODEC Module Registers

TSS Register (TSSR)

The TSS Register (TSSR) corresponds to the `gdTSSTransmitter` protocol parameter. This parameter is a global cluster parameter that indicates the number of bits (`gdBit`) in the Transmit Start Sequence.

The TSSR accepts values from 3 to 15 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 40 shows the bit positions in the TSSR and Table 41 provides the TSSR description .

Table 40: **TSS Register Bit Positions**

0 - 27	28 - 31
RESERVED	GTSSTX(3:0)

Table 41: **TSS Register Description**

Bit(s)	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	GTSSTX(3:0)	Read/Write	8	**gdTSSTransmitter**: The number of gdBits bits that make up the Transmission Start Sequence (TSS).

WUP Receive Idle Register (WRIR)

Because of the collisions on the bus and clock differences, the Wakeup Symbol decoded by a node may be shorter than the transmitted Wakeup Symbol. The WUP Receive Idle Register (WRIR) corresponds to the `gdWakeupSymbolRxIdle` protocol parameter. It is a global cluster parameter that indicates the number of bits (`gdBit`) that must be decoded to detect the 'idle' portion of the Wakeup Symbol.

The WRIR accepts values from 14 to 59 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 42 shows the bit positions in WRIR and Table 43 provides the WRIR description.

Table 42: **WUP Receive Idle Register Bit Positions**

0-25	26-31
RESERVED	GWSRXIDLE(5:0)

Table 43: **WUP Receive Idle Register Description**

Bit(s)	Name	Access	Default Value	Descripition
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
26-31	GWSRXIDLE(5:0)	Read/Write	20	**gdWakeupSymbolRxIdle**: Indicates the number of bits (gdBit) required to detect the 'idle' part of the wakeup symbol.

WUP Receive Low Register (WRLR)

Because of the collisions on the bus and clock differences, the Wakeup Symbol decoded by a node may be shorter than the transmitted Wakeup Symbol. The WUP Receive Low Register (WRLR) corresponds to the `gdWakeupSymbolRxLow` protocol parameter. It is a global cluster parameter that indicates the number of bits (`gdBit`) that must be decoded to detect the 'low' portion of the Wakeup Symbol.

The WRLR accepts values from 11 to 59 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 44 shows the bit positions in the WRLR and Table 45 provides the WRLR description.

Table 44: **WUP Receive Low Register Bit Positions**

0-24	26-31
RESERVED	GWSRXLOW(5:0)

Table 45: **WUP Receive Low Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
26-31	GWSRXLOW(5:0)	Read/Write	20	**gdWakeupSymbolRxLow**: The number of bits (gdBit) required to detect the 'low' part of the wakeup symbol.

WUP Receive Window Register (WRWR)

Because of the collisions on the bus and clock differences, the Wakeup Symbol decoded by a node may be longer than the transmitted Wakeup Symbol. The WUP Receive Window Register (WRWR) corresponds to the `gdWakeupSymbolRxWindow` protocol parameter. It is a global cluster parameter that indicates the number of bits (`gdBit`) that must be decoded to detect the 'low' and 'idle' portion of the first Wakeup Symbol, and the 'low' portion of the following wakeup symbol.

The WRWR accepts values from 76 to 301 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 46 shows the bit positions in the WRWR and Table 47 provides the WRWR description.

Table 46: **WUP Receive Window Register Bit Positions**

0-22	23-31
RESERVED	GWSRXWIN(8:0)

Table 47: **WUP Receive Window Register Description**

Bits	Name	Access	Default Value	Description
0-22	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
23-31	GWSRXWIN(8:0)	Read	160	**gdWakeupSymbolRxWindow**: Indicates the maximum wakeup duration window with in which two 'low' portions and one 'idle' portion of the wakeup symbol must be received.

WUP Transmit Idle Register (WTIR)

The WUP Transmit Idle Register (WTIR) corresponds to the `gdWakeupSymbolTxIdle` protocol parameter. It is a global cluster parameter indicates the length (`gdBit`) of the 'idle' portion of the wakeup symbol transmitted by the controller.

The WTIR accepts values from 45 to 180 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 48 shows the bit positions in the WTIR and Table 49 provides the WTIR description.

Table 48: **WUP Transmit Idle Register Bit Positions**

0-23	24-31
RESERVED	GWSTXIDLE(7:0)

Table 49: **WUP Transmit Idle Register Description**

Bits	Name	Access	Default Value	Description
0-23	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
24-31	GWSTXIDLE(7:0)	Read/Write	60	**gdWakeupSymbolTxIdle:** Indicates the number of bits in the idle portion of the wakeup symbol transmitted by the controller.

WUP Transmit Low Register (WTLR)

The WUP Transmit Low Register (WTLR) corresponds to the `gdWakeupSymbolTxLow` protocol parameter. It is a global cluster parameter indicates the length (`gdBit`) of the 'low' portion of the wakeup symbol that is transmitted by the controller.

The WTLR accepts values from15 to 60 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 50 shows the bit positions in the WTLR and Table 51 provides the WTLR description.

Table 50: **WUP Transmit Low Register Bit Positions**

0-25	26-31
RESERVED	GWSTXLOW(5:0)

Table 51: **WUP Transmit Low Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
26-31	GWSTXLOW(5:0)	Read/Write	40	**gdWakeupSymbolTxLow**: Indicates the number of bits in the low portion of the wakeup symbol transmitted by the controller.

CAS Receive Max Register (CRMXR)

The CAS Receive Max Register (CRMXR) corresponds to the `gdCASRxLowMax` protocol parameter. This a global cluster parameter that indicates the maximum number of bits that need to be detected 'low' to detect a CAS symbol.

The CRMXR accepts values from 67 to 99 `gdBit`.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 52 shows the bit positions in the CRMXR and Table 53 provides the CRMXR description.

Table 52: **CAS Receive Max Register Bit Positions**

0-24	25-31
RESERVED	GCRLM(6:0)

Table 53: **CAS Receive Max Register Description**

Bits	Name	Access	Default Value	Description
0-24	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
25-31	GCRLM(6:0)	Read/Write	75	**gdCASRxLowMax**: Maximum number of bits that need to be detected 'low' to detect a CAS symbol.

Decoding Correction Register (DCR)

The Decoding Correction Register (DCR) indicates the summation of the protocol parameters `pDecodingCorrection` and `pDelayCompensation`. The DCR indicates the time difference value between the primary and secondary time reference points.

The DCR accepts values from 14 to 343.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 54 shows the bit positions in the DCR and Table 55 provides the DCR description.

Table 54: **Decoding Correction Register Bit Positions**

0-23	24-31
RESERVED	DECCORR(7:0)

Table 55: **Decoding Correction Register Description**

Bits	Name	Access	Default Value	Description
0-23	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
24-31	DECCORR(7:0)	Read/Write	51	**pDecodingCorrection** and **pDelayCompensation**: The value (microticks) used to calculate the difference between primary reference point and secondary reference point.

Wakeup Pattern Sequence Register (WPSR)

The Wakeup Pattern Sequence Register (WPSR) corresponds to the `pWakeupPattern` protocol parameter. This parameter is node-specific and indicates the number of times a Wakeup Symbol is transmitted by the FlexRay controller to form the wakeup pattern.

The WPSR accepts values from 2 to 63.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 56 shows the bit positions in the WPSR and Table 57 provides the WPSR description.

Table 56: **Wakeup Pattern Sequence Register Bit Positions**

0-25	26-31
RESERVED	PWUPP(5:0)

Table 57: **Wakeup Pattern Sequence Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
26-31	PWUPP(5:0)	Read/Write	2	**pWakeupPattern**: Indicates the number of repetitions of the wakeup symbol to form the wakeup pattern

MAC Module Registers

Sync Slot Register (SYSR)

The Sync Slot Register (SYSR) corresponds to the `pKeySlotId` protocol parameter. This parameter is node-specific and indicates the slot during which a sync frame/startup frame is sent.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 58 shows the bit positions in the SYSR and Table 59 provides the SYSR description.

Table 58: **Sync Slot Register Bit Positions**

0-21	22-31
RESERVED	PKEYSLOTID(9:0)

Table 59: **Sync Slot Register Description**

Bits	Name	Access	Default Value	Description
0-21	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
22-31	PKEYSLOTID(9:0)	Read/Write	0	**pKeySlotId**: Indicates the slot during which a sync frame/startup frame is sent.

MiniSlot Duration Register (MSDR)

The MiniSlot Duration Register MSDR corresponds to the `gdMiniSlot` protocol indicator. This is a global cluster parameter that indicates the duration of a minislot (in macroticks).

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The MSDR accepts values from 2 to 63 macroticks.

Table 60 shows the bit positions in the MSDR and Table 61 provides the MSDR description.

Table 60: **MiniSlot Duration Register Bit Positions**

0-25	26-31
RESERVED	GDMINISLOT(5:0)

Table 61: **MiniSlot Duration Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
26-31	GDMINISLOT(5:0)	Read/Write	37 Macroticks	**gdMiniSlot**: Indicates the duration of minislot in macroticks

MiniSlot Action Point Offset Register (MAPR)

The MiniSlot Action Point Offset Register (MAPR) corresponds to the `gdMiniSlotActionPointOffset` protocol parameter. This is a global cluster parameter that indicates the number of macroticks the minislot action point is offset from the beginning of a minislot.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The MAPR accepts values from 1 to 31 macroticks.

Table 62 shows the bit positions in the MAPR and Table 63 provides the MAPR description.

Table 62: **MiniSlot Action Point Offset Register Bit Positions**

0-26	27-31
RESERVED	GMSAPO(4:0)

Table 63: **MiniSlot Action Point Offset Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
27-31	GMSAPO(4:0)	Read/Write	2	**gdMiniSlotActionPointOffset**: Indicates the number of macroticks the minislot action point is offset from the start of a minislot

Dynamic Slot Idle Phase Register (DSIPR)

The Dynamic Slot Idle Phase Register (DSIPR) corresponds to the `gdDynamicSlotIdlePhase` protocol parameter. This is a global cluster parameter that indicates the number of minislots in the idle phase within a dynamic slot.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Write—Allowed only in Config state.

The DSIPR accepts values from 0 to 2 minislots.

Table 64 shows the bit positions in the DSIPR and Table 65 provides the DSIPR description.

Table 64: **Dynamic Slot Idle Phase Register Bit Positions**

0-29	30-31
RESERVED	GDSIP(1:0)

Table 65: **Dynamic Slot Idle Phase Register Description**

Bits	Name	Access	Default Value	Description
0-29	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
30-31	GDSIP(1:0)	Read/Write	1	**gdDynamicSlotIdlePhase**: Indicates the number of minislots that constitute the idle phase in a dynamic slot.

Number Of MiniSlots Register (NMSR)

The Number Of MiniSlots Register (NMSR) corresponds to the `gNumberOfMiniSlots` protocol parameter. This is a global cluster parameter that indicates the number of minislots present in the dynamic segment.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The DSIPR accepts values from 0 to 7986 minislots.

Table 66 shows the bit positions in the NMSR and Table 67 provides the NMSR description.

Table 66: **Number Of MiniSlots Register Bit Positions**

0-18	19-31
RESERVED	GNOMS(12:0)

Table 67: **Number Of MiniSlots Register Description**

Bits	Name	Access	Default Value	Description
0-18	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
19-31	GNOMS(12:0)	Read/Write	18 minislots	**gNumberOfMiniSlots**: Indicates the number of minislots in dynamic segment

Latest Tx Register (LTR)

In the dynamic segment, frames can be transmitted only in mini slots that do not exceed the `pLatestTx` minislot. The Latest Tx Register (LTR) corresponds to the `pLatestTx` protocol parameter. This is a node-specific cluster parameter that indicates the last minislot in the dynamic segment during which a frame transmission can start.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The LTR accepts values from 0 to 7980 minislots.

Table 68 shows the bit positions in the LTR and Table 69 provides the LTR description.

Table 68: **Latest Tx Register Bit Positions**

0-18	19-31
RESERVED	PLATESTTX(12:0)

Table 69: **Latest Tx Register Description**

Bits	Name	Access	Default Value	Description
0-18	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
19-31	PLATESTTX(12:0)	Read/Write	17	**pLatestTx**: Indicates the last minislot in the dynamic segment during which frame transmission can start.

Symbol Window Duration Register (SWDR)

The Symbol Window Duration Register (SWDR) corresponds to the `gdSymbolWindow` protocol parameter. This is a global cluster parameter that indicates the duration of the symbol window in terms of macroticks.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The SWDR accepts values from 0 to 142 macroticks.

Table 70 shows the bit positions in the SWDR and Table 71 provides the SWDR description.

Table 70: **Symbol Window Duration Register Bit Positions**

0-23	24-31
RESERVED	GSW(7:0)

Table 71: **Symbol Window Duration Register Description**

Bits	Name	Access	Default Value	Description
0-23	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
24-31	GSW(7:0)	Read/Write	69	**gdSymbolWindow**: Indicates the duration of the Symbol Window in Macroticks. A value of 0 indicates the absence of the Symbol Window in the communication cycle.

MTS Transmit Register (MTR)

During Normal Active and Normal Passive operational modes, the Media Test Access (MTS) symbol can be transmitted during the Symbol Window. For cluster configurations where in the Symbol window exists, the MTS Transmit Register (MTR) governs the transmission of symbols.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in the Normal Active and Normal Passive states.

Table 72 shows the bit positions in the MTR and Table 73 provides the MTR description.

Table 72: **MTS Transmit Register Bit Positions**

0-30	31
RESERVED	SSM

Table 73: **MTS Transmit Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	SSM	Read/Write	0	**Shall Send MTS:** This bit governs the transmission of the MTS Symbol during the Symbol Window. '1' = The MAC module in the PE sends an MTS in the symbol window. '0' = The PE does not send an MTS during the symbol window. The SSM bit is cleared by the controller after the transmission of a MTS symbol. The SSM bit should be set to '1' before the start of the symbol window.

Slot Counter Status Register (SCSR)

The Slot Counter Status Register (SCSR) corresponds to the `vSlotCounter` protocol status parameter. It is a protocol status parameter that indicates the current slot number in the communication cycle. `vSlotCounter` is initialized to '1' at the start of each communication cycle and is incremented at the end of each slot. This continues until one of the following conditions is met:

- The slot counter has reached `cSlotIDMax` (2047), or
- The current slot in the dynamic segment has reached the minislot `gNumberOfMinislots`; for example, the end of the dynamic segment.

When one of these conditions is met, `vSlotCounter` is set to '0' for the remainder of the communication cycle.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 74 shows the bit positions in the SCSR and Table 75 provides the SCSR description.

Table 74: **Slot Counter Status Register Bit Positions**

0-20	21-31
RESERVED	VSLOTCOUNT(10:0)

Table 75: **Slot Counter Status Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
21-31	VSLOTCOUNT(10:0)	Read	0	**vSlotCounter**: Indicates the current slot number in the communication cycle. At the end of every communication cycle, this value is cleared to '0.'

Last Dynamic Frame Transmit Status Register (LDFTSR)

This Last Dynamic Frame Transmit Status Register (LDFTSR) corresponds to the `zLastDynTxSlot` protocol status parameter. This parameter is a protocol status parameter that indicates slot number of the last transmitted frame in the dynamic segment.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 76 shows the bit positions in the LDFTSR and Table 77 provides the LDFTSR description.

Table 76: **Last Dynamic Frame Transmit Status Register Bit Positions**

0-20	21-31
RESERVED	ZLDTXS(10:0)

Table 77: **Last Dynamic Frame Transmit Status Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
21-31	ZLDTXS(10:0)	Read	0	**zLastDynTxSlot**: Indicates the slot number during which last dynamic frame was transmitted. At the beginning of dynamic segment, this bit field is reset to '0.'.

Clock Sync Module Registers

Extern Offset Correction Register (EOCR)

The Extern Offset Correction Register (EOCR) corresponds to the `pExternOffsetCorrection` protocol parameter. This parameter is a node-specific protocol parameter that indicates the number of microticks that are added or subtracted from the NIT during external offset correction.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The EOCR accepts values from 0 to 7 microticks.

Table 78 shows the bit positions in the EOCR and Table 79 provides the EOCR description.

Table 78: **Extern Offset Correction Register Bit Positions**

0-28	29-31
RESERVED	PEXTOCOR(2:0)

Table 79: **Extern Offset Correction Register Description**

Bits	Name	Access	Default Value	Description
0-28	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
29-31	PEXTOCOR(2:0)	Read/Write	0	**pExternOffsetCorrection**: Indicates the number of microticks that are added or subtracted from the NIT to carry out external offset correction.

Extern Rate Correction Register (ERCR)

The Extern Rate Correction Register (ERCR) corresponds to the `pExternRateCorrection` protocol parameter. This is a node-specific protocol parameter that indicates the number of microticks that are added or subtracted from the NIT to carry out external rate correction.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The ERCR accepts values from 0 to 7 microticks.

Table 80 shows the bit positions in the ERCR and Table 81 provides the ERCR description.

Table 80: **Extern Rate Correction Register Bit Positions**

0-28	29-31
RESERVED	PEXTRCOR(2:0)

Table 81: **Extern Rate Correction Register Description**

Bits	Name	Access	Default Value	Description
0-28	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
29-31	PEXTRCOR(2:0)	Read/Write	0	**pExternRateCorrection**: Indicates the number of microticks that are added or subtracted from the NIT to carry out external rate correction.

Apply Extern Correction Register (AECR)

The Apply Extern Correction Register (AECR) controls the application of the values specified in the ERCR and EOCR to perform external rate/offset correction.

Access Criterion for Reads/Writes

- Read—Allowed in all operational states.
- Writes—Allowed in the Normal Active state of operation.

Table 82 shows the bit positions in the AECR and Table 83 provides the AECR description.

Table 82: **Apply Extern Correction Register Bit Positions**

0-27	28-29	30-31
Reserved	AEOCORR(1:0)	AERCORR(1:0)

Table 83: **Apply Extern Correction Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-29	AEOCORR(1:0)	Read/Write	0	**Apply Extern Offset Correction**: Controls the application of the value indicated by the EOCR. '01' - Subtract the value indicated by the EOCR to perform External Offset Correction. '10' - Add the value indicated by the EOCR to perform External Offset Correction. '11' - Do not perform External Offset Correction.
30-31	AERCORR(1:0)	Read/Write	0	**Apply Extern Rate Correction**: Controls the application of the value indicated by the ERCR. '01' - Subtract the value indicated by the ERCR to perform External Rate Correction. '10' - Add the value indicated by the ERCR to perform External Rate Correction. '11' - Do not perform External Rate Correction.

Offset Correction Limit Register (OCLR)

The Offset Correction Limit Register (OCLR) corresponds to the `pOffsetCorrectionOut` protocol parameter. This is a node-specific protocol parameter that indicates the maximum offset correction value.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The OCLR accepts values from 13 to 15567 microticks.

Table 84 shows the bit positions in the OCLR and Table 85 provides the OCLR description.

Table 84: **Offset Correction Limit Register Bit Positions**

0-17	18-31
RESERVED	POCORO(13:0)

Table 85: **Offset Correction Limit Register Description**

Bits	Name	Access	Default Value	Description
0-17	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
18-31	POCORO(13:0)	Read/Write	1	**pOffsetCorrectionOut**: Indicates the maximum permissible offset correction.

Rate Correction Limit Register (RCLR)

The Rate Correction Limit Register (RCLR) corresponds to the `pRateCorrectionOut` protocol parameter. This is a node-specific protocol parameter that indicates the maximum rate correction value.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The RCLR accepts values from 2 to 1923 microticks.

Table 86 shows the bit positions in the RCLR and Table 87 provides the RCLR description.

Table 86: **Rate Correction Limit Register Bit Positions**

0-20	21-31
RESERVED	PRCORO(10:0)

Table 87: **Rate Correction Limit Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
21-31	PRCORO(10:0)	Read/Write	1	**pRateCorrectionOut**: Indicates the maximum permissible rate correction.

Number Of Microticks Per Cycle Register (NMICR)

The Number Of Microticks Per Cycle Register (NMICR) corresponds to the `pMicroPerCycle` protocol parameter. This is a node-specific protocol parameter that indicates the number of microticks in the communication cycle.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The NMICR accepts values from 640 to 640000 microticks.

Table 88 shows the bit positions in the NMICR and Table 89 provides the NMICR description.

Table 88: **Number Of Microticks Per Cycle Register Bit Positions**

0-11	12-31
RESERVED	PMIPC(19:0)

Table 89: **Number Of Microticks Per Cycle Register Description**

Bits	Name	Access	Default Value	Description
0-11	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
12-31	PMIPC(19:0)	Read/Write	60000	**pMicroPerCycle**: Indicates the number of microticks in a communication cycle.

Number Of Macroticks Per Cycle Register (NMACR)

The Number Of Macroticks Per Cycle Register (NMACR) corresponds to the `gMacroPerCycle` protocol parameter. This is a global cluster parameter that indicates the number of macroticks in the communication cycle.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The NMACR accepts values from 10 to 16000 macroticks.

Table 90 shows the bit positions in the NMACR and Table 91 provides the NMACR description.

Table 90: **Number of Macroticks Per Cycle Register Bit Positions**

0-17	18-31
RESERVED	GMAPC(13:0)

Table 91: **Number of Macroticks Per Cycle Register Description**

Bits	Name	Access	Default Value	Description
0-17	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
18-31	GMAPC(13:0)	Read/Write	1500	**gMacroPerCycle**: Indicates the number of macroticks in a communication cycle.

Maximum Drift Register (MDR)

The Maximum Drift Register (MDR) corresponds to the `pdMaxDrift` protocol parameter. This is a node-specific protocol parameter that indicates maximum drift offset between two nodes that are operating with unsynchronized clocks over one communication cycle.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The MDR accepts values from 2 to 1923 microticks.

Table 92 shows the bit position in MDR and Table 93 provides the MDR description.

Table 92: **Maximum Drift Register Bit Positions**

0-20	21-31
RESERVED	PDMD(10:0)

Table 93: **Maximum Drift Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
21-31	PDMD(10:0)	Read/Write	2	**pdMaxDrift**: Indicates maximum drift offset between two nodes operating with unsynchronized clocks over one communication cycle.

Cluster Drift Damping Register (CDDR)

The Cluster Drift Damping Register (CDDR) corresponds to the `pClusterDriftDamping` protocol parameter. This is a node-specific protocol parameter that indicates the cluster drift damping value used for rate correction.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The CDDR accepts values from 0 to 20 microticks.

Table 94 shows the bit positions in the CDDR and Table 95 provides the CDDR description.

Table 94: **Cluster Drift Damping Register Bit Positions**

0-26	27-31
RESERVED	PCDDAMP(4:0)

Table 95: **Cluster Drift Damping Register Bit Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
27-31	PCDDAMP(4:0)	Read/Write	0	**pClusterDriftDamping**: Indicates the cluster drift damping value used for rate correction

Macro Initial Offset Register (MAOR)

The Macro Initial Offset Register (MAOR) corresponds to the `pMacroInitialOffset` protocol parameter. This is a node-specific protocol parameter that indicates the number of macroticks between the static slot boundary and the following macrotick boundary of the secondary time reference point.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The MAOR accepts values from 2 to 68 microticks.

Table 96 shows the bit positions in the MAOR and Table 97 provides the MAOR description.

Table 96: **Macro Initial Offset Register Bit Positions**

0-24	25-31
RESERVED	PMAIO(6:0)

Table 97: **Macro Initial Offset Register Description**

Bits	Name	Access	Default Value	Description
0-24	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
25-31	PMAIO(6:0)	Read/Write	4	**pMacroInitialOffset**: Number of macroticks between the static slot boundary and the subsequent macrotick boundary of the secondary time reference point.

Offset Correction Start Register (OCSR)

The Offset Correction Start Register (OCSR) corresponds to the `gOffsetCorrectionStart` protocol parameter. This is a global cluster parameter that indicates the number of macroticks between the start of the communication cycle and the start of the offset correction within the NIT.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The OCSR accepts values from 9 to 15999 macroticks.

Table 98 shows the bit positions in the OCSR and Table 99 provides the OCSR description.

Table 98: **Offset Correction Start Register Bit Positions**

0-17	18-31
RESERVED	GOCORS(13:0)

Table 99: **Offset Correction Start Register Description**

Bits	Name	Access	Default Value	Description
0-19	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
18-31	GOCORS(13:0)	Read/Write	15600	**gOffsetCorrectionStart**: Indicates the number of macroticks between the start of the communication cycle and the start of the offset correction within the NIT

Micro Initial Offset Register (MIOR)

The Micro Initial Offset Register (MIO) corresponds to the `pMicroInitialOffset` protocol parameter. This is a node-specific protocol parameter that indicates the number of microticks between the closest macrotick boundary described by the MAOR and the secondary time reference point.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The MIOR accepts values from 0 to 239 microticks.

Table 100 shows the bit positions in the MIOR and Table 101 provides the MIOR description.

Table 100: **Micro Initial Offset Register Bit Positions**

0-23	24-31
RESERVED	PMIIO(7:0)

Table 101: **Micro Initial Offset Register Description**

Bits	Name	Access	Default Value	Description
0-23	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
24-31	PMIIO(7:0)	Read/Write	23	**pMicroInitialOffset**: Indicates the number of microticks between the closest macrotick boundary described by the MAOR and the secondary time reference point.

Sync Node Max Register (SNMR)

The Sync Node Max Register (SNMR) corresponds to the `gSyncNodeMax` protocol parameter. This is a global cluster parameter that indicates the maximum number of nodes in the cluster that can send sync frames.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The SNMR accepts values from 2 to 15.

Table 102 shows the bit positions in the SNMR and Table 103 provides the SNMR description.

Table 102: **Sync Node Max Register Bit Positions**

0-27	28-31
RESERVED	GSNM(3:0)

Table 103: **Sync Node Max Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
28-31	GSNM(3:0)	Read/Write	2	**gSyncNodeMax**: Indicates the number of nodes in the cluster than can send sync frames.

Accepted Startup Range Register (ASRR)

The Accepted Startup Range Register (ASRR) corresponds to the `pdAcceptedStartupRange` protocol parameter. This is a node-specific parameter that indicates the maximum range of measured clock deviation allowed for startup frames during integration stage.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The ASRR accepts values from 0 to 1875 microticks.

Table 104 shows the bit positions in the ASRR and Table 105 provides the ASRR description.

Table 104: **Accepted Startup Range Register Bit Positions**

0-20	21-31
RESERVED	PASR(10:0)

Table 105: **Accepted Startup Range Register Description**

Bits	Name	Access	Default Value	Description
0-20	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
21-31	PASR(10:0)	Read/Write	0	**pdAcceptedStartupRange**: Indicates the maximum range of measured clock deviation allowed for startup frames during integration stage.

Rate Correction Status Register (RCSR)

The Rate Correction Status Register (RCSR) corresponds to the `vRateCorrection` protocol parameter. This is a node-specific protocol status parameter that indicates the rate correction value computed by the clock synchronization module.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 106 shows the bit positions in the RCSR and Table 107 provides the RCSR description.

Table 106: **Rate Correction Status Register Bit Position**

0-19	20-31
RESERVED	VRCOR(11:0)

Table 107: **Rate Correction Status Register Bit Field Description**

Bits	Name	Access	Default Value	Description
0-19	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
20-31	VRCOR(11:0)	Read	0	**vRateCorrection**: Indicates the current rate correction value. Indicated in 2's complement format.

Offset Correction Status Register (OFCSR)

The Offset Correction Status Register (OFCSR) corresponds to the `vOffsetCorrection` protocol parameter. This is a node-specific protocol status parameter that indicates the offset correction value computed by the clock synchronization module.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 108 shows the bit positions in the OFCSR and Table 109 provides the OFCSR description.

Table 108: **Offset Correction Status Register Bit Positions**

0-17	18-31
RESERVED	VOCOR(13:0)

Table 109: **Offset Correction Status Register Description**

Bits	Name	Access	Default Value	Description
0-17	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
18-31	VOCOR(13:0)	Read	0	**vOffsetCorrection**: Indicates the current offset correction value. Indicated in 2's complement format.

Valid Sync Frames Status Register (VSFSR)

The Valid Sync Frames Status Register (VSFSR) indicates the number of valid entries in the SFSR[1..30] registers.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 110 shows the bit positions in the VSFSR and Table 111 provides the VSFSR description.

Table 110: **Valid Sync Frames Status Register Bit Positions**

0-22	23	24-27	28-31
RESERVED	SFSRUP: SFSR[1..30]	NUMSFODD[3..0]	NUMSFEVEN[3:0]

Table 111: **Valid Sync Frames Status Register Description**

Bits	Name	Access	Default Value	Description
0-22	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
23	SFSRUP:SFSR[1..30]	Read	0	**Update bit**: Indicates that the SFSR[1..30] registers are being updated. '1' = Indicates that SFSR[1..30] cannot be read by the host. '0' = Indicates that SFSR[1..30] can be read by the host.
24-27	NUMSFODD[3..0]	Read	0	**Number of Odd Sync Frames**: Indicates the number of valid entries in the SFSR[16..30] registers.
28-31	NUMSFEVEN[3:0]	Read		**Number of Even Sync Frames**: Indicates the number of valid entries in the SFSR[1..15] Registers.

Cycle Count Status Register (CCSR)

The Cycle Count Status Register (CCSR) corresponds to the `vCycleCounter` protocol parameter. This is a node-specific protocol status parameter that indicates the current communication cycle number.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 112 shows the bit positions in the CCSR and Table 113 provides the CCSR description.

Table 112: **Cycle Count Status Register Bit Positions**

0-25	26-31
RESERVED	VCYCLECNT(5:0)

Table 113: **Cycle Count Status Register Description**

Bits	Name	Access	Default Value	Description
0-25	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
26-31	VCYCLECNT(5:0)	Read	0	**vCycleCounter**: Indicates the current communication cycle number

Macrotick Status Register (MSR)

The Macrotick Status Register (MSR) corresponds to the `vMacrotick` protocol parameter. This is a node-specific protocol status parameter that indicates the current value of the macrotick.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

The MSR indicates values from 0 to (gMacroPerCycle -1). `gMacroPerCycle` is indicated by the Number Of Macroticks Per Cycle register.

Table 114 shows the bit positions in the MSR and Table 115 provides the MSR description.

Table 114: **Macrotick Status Register Bit Positions**

0-17	18-31
RESERVED	VMCTICK(13:0)

Table 115: **Macrotick Status Register Description**

Bits	Name	Access	Default Value	Description
0-17	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
18-31	VMCTICK(13:0)	Read	0	**vMacrotick**: Indicates the current value of the Macrotick.

Sync Frames Status Registers [1..30] (SFSR[n])

The Sync Frames Status Registers [1..30] (SFSR[n]) corresponds to the `vsSyncIdList` protocol parameter. This is a node-specific protocol status parameter that indicates the Slot IDs of all the sync frames received during the communication cycle.

The first set of 15 registers hold the Slot IDs of sync frames received during an even communication cycle. The second set of 15 registers hold the Slot IDs of sync frames transmitted or received during an odd communication cycle.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states except when the SFSRUP bit in VSFSR is '1'.
- Writes—Not allowed.

Table 116 shows the bit positions in the SFSR[n] and Table 117 provides the SFSR[n] description.

Table 116: **Sync Frames Status Register[n] Bit Position**

0-21	22-31
RESERVED	SYNCIDLST(9:0)

Table 117: **Sync Frames Status Register[n] Bit Field Description**

Bits	Name	Access	Default Value	Description
0-21	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
22-31	SYNCIDLST(9:0)	Read	0	**vsSyncIdList**: Indicates the Slot ID during which a sync frame was transmitted or received.

Wakeup Module Registers

Listen Timeout Register (LTOR)

The Listen Time-out Register (LTOR) corresponds to the pdListenTimeout protocol parameter. This is a node-specific cluster parameter that indicates the upper limit for startup listen time-out and wake up listen time-out in the absence of noise.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The LTOR accepts values from 1284 to 1283846 microticks.

Table 118 shows the bits position in the LTOR and Table 119 provides the LTOR description

Table 118: **Listen Timeout Register Bit Positions**

0-10	11-31
RESERVED	PLTO(20:0)

Table 119: **Listen Timeout Register Description**

Bits	Name	Access	Default Value	Description
0-10	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
11-31	PLTO(20:0)	Read/Write	2000	**pdListenTimeout**: Indicates the upper limit for startup listen time-out and wake up listen time-out.

Listen Noise Register (LNR)

Bits 27 to 31 of the LNR corresponds to the protocol parameter gListenNoise. This is a global cluster parameter that when multiplied with the value indicated by LTOR (pdListenTimeout) indicates the upper limit for startup listen timeout and wake up listen timeout in the presence of noise. Bits 1 to 26 of the LNR correspond to this product gListenNoise*pdListenTimeout.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

GLN takes values from 2 to 16. LNP takes values from 2568 to 20541536.

Table 120 shows the bits position in the LNR and Table 121 provides the LNR description

Table 120: **Listen Noise Register Bit Positions**

0	1-26	27-31
RESERVED	LNP(25:0)	GLN(4:0)

Table 121: **Listen Noise Register Description**

Bits	Name	Access	Default Value	Description
0	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
1-26	LNP(25:0)	Read/Write	8000	**gListenNoise*pdListenTimeout:** This product indicates the upper limit for startup listen timeout and wakeup listen timeout in the presence of noise.
27	GLN(4:0)	Read/Write	4	**gListenNoise:** Indicates the multiplication factor for the global cluster parameter pdListenTimeOut.

Number Of ColdStart Attempts Register (NCAR)

The Number Of Cold Start Attempts Register (NCAR) corresponds to the `gColdStartAttempts` protocol parameter. This is a global cluster parameter that indicates the maximum number of times a node in the cluster is permitted to attempt to start the cluster by initiating schedule synchronization.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

The NCAR accepts values from 2 to 31.

Table 122 shows the bit positions in the NCAR and Table 123 provides the NCAR description.

Table 122: **Number Of Coldstart Attempts Register Bit Positions**

0-26	27-31
RESERVED	GCSA(4:0)

Table 123: **Number Of Coldstart Attempts Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
27-31	GCSA(4:0)	Read/Write	2	**gColdStartAttempts**: Indicates the maximum number of times a node is allowed to attempt to start the cluster

Wakeup Status Register (WSR)

The Wakeup Status Register (WSR) contains POC status information, and defines the `vPOC!WakeupStatus` status variable, as specified in the *FlexRay Protocol Specification Rev 2.1 A*.

Table 124 shows the bit positions in the WSR and Table 125 provides the WSR description.

Table 124: **Wakeup Status Register Bit Positions**

0-28	29-31
RESERVED	WUPS(2:0)

Table 125: **Wakeup Status Register Description**

Bits	Name	Access	Default Value	Description
0-24	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
25	WUPS(2:0)	Read	0	**vPOC!WakeupStatus**: Indicates the outcome of the wakeup operation. ‘00’ = UNDEFINED ’000’ = Reserved ’001’ = Reserved ’010’ = Indicates RECEIVED_HEADER ’011’ = Indicates RECEIVED_WUP ’100’ = Indicates COLLISION_HEADER ’101’ = Indicates COLLISION_WUP ’110’ = Indicates COLLISION_UNKNOWN ’111’ = Indicates TRANSMITTED

Coldstart Noise Status Register (CNSR)

The Cold Start Noise Status Register (CNR) corresponds to the `vPOC!ColdstartNoise` protocol status variable. This variable indicates whether the noise was observed during the cold start operation.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writing a'1' to this bit clears it.

Table 126 shows the bit positions in the CNSR and Table 127 provides the CNSR description.

Table 126: **Coldstart Noise Register Status Bit Positions**

0-30	31
RESERVED	CSNOISE

Table 127: **Coldstart Noise Status Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
31	CSNOISE	Read/Write	0	**vPOC!ColdstartNoise**: Indicates the noise status during a cold start operation. '1' = Indicates that noise was observed during the cold start operation. '0' = Indicates that noise was not observed during the cold start operation. Writing a '1' to this bit clears it.

Remaining Coldstart Attempts Status Register (RCASR)

The Remaining Cold Start Attempts Status Register (RCASR) corresponds to the `vRemainingColdStartAttempts` protocol status variable. The number of cold start attempts made by the node is initially indicated by the NCAR (`gColdStartAttemps`); for every attempt to cold start, the number of pending attempts is reduced by 1. `vRemainingColdStartAttempts` indicates the number of pending cold start attempts.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Writes are not allowed.

Table 128 shows the bit positions in the RCASR and Table 129 provides the RCASR description.

Table 128: **Remaining Coldstart Attempts Status Register Bit Positions**

0-26	27-31
RESERVED	VRCSATTEMPTS(4:0)

Table 129: **Remaining Coldstart Attempts Status Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
27-31	VRCSATTEMPTS(4:0)	Read/Write	2	**vRemainingColdStartAttempts**: Indicates the number of pending cold start attempts.

Startup Status Register (STSR)

The Startup Status Register (STSR) contains POC status information, and defines `vPOC!StartupState` status variable, as specified in the *FlexRay Protocol Specification Rev 2.1 A*.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 130 shows the bit positions in the STSR and Table 131 provides the STSR description.

Table 130: **Startup Status Register Bit Positions**

0-27	28-31
RESERVED	SUPST(3:0)

Table 131: **Startup Status Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
28-31	SUPST(3:0)	Read	0	**vPOC!StartupState**: '0000' = Reserved. '0001' = Indicates COLDSTART_LISTEN state. '0010' = Indicates INTEGRATION_COLDSTART_ CHECK state. '0011' = Indicates CODLSTART_ JOIN state. '0100' = Indicates COLDSTART_COLLISION_ RESOLUTION state. '0101' = Indicates COLDSTART_CONSISTENCY_ CHECK state. '0110' = Indicates INTEGERATION_LISTEN state. '0111' = Indicates INITIALIZE_SCHEDULE state. '1000' = Indicates INTEGRATION_CONSISTENCY_ CHECK state. '1001' = Indicates COLDSTART_GAP state. '1010' = UNDEFINED '1011 to '1111' = Reserved.

Timers

The FlexRay controller support two types of timers: Absolute and Relative. These timers work when the controller is in Normal Active or Normal Passive states. When each of the timers expire they set bits in the CISR, and the `Flex_Cirpt` line can be configured to be asserted.

The Absolute timer can be disabled by writing a '0' to the ATEN bit in the ATCR. The Relative timer can be disabled by writing a '0' to the RTEN bit in the RTCR.

Absolute Timer Control Register (ATCR)

The Absolute Timer Control Register (ATCR) contains the enable bit for the Absolute timer. The enable bit is cleared to a '0' by the controller if the operational mode of the timer is non-repetitive. Writing a '0' to the enable bit disables the Absolute timer.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in the Normal Active or Normal Passive operational state.

Table 132 shows the bit positions in the ATCR and Table 133 provides the ATCR description.

Table 132: **Absolute Timer Control Register Bit Positions**

0-30	31
RESERVED	ATEN

Table 133: **Absolute Timer Control Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	ATEN	Read/Write	0	**Absolute Timer Enable**: The ATEN bit acts as an enable for the Absolute Timer. '1' = Writing a '1' to this bit enables the Absolute timer. '0' = Writing a '0' to this bit disables the Absolute timer. Reads to this bit, indicate whether the Absolute timer is enabled or disabled. The ATEN bit is cleared to a '0' by the FlexRay controller after the timer elapses only when the ATMODE bit in the ATMOR is a '0.'

Absolute Timer Cycle Count Register (ATCCR)

The Absolute Timer Cycle Count Register (ATCCR) contains the Cycle ID and Cycle ID Mask fields.

Access Criterion for Reads/Writes

- Read—Allowed in all operational states.
- Writes—Allowed in the Normal Active or Normal Passive operational state only when the ATEN bit in the ATCR is a '0.'

Table 134 shows the bit positions in the ATCCR and Table 135 provides the ATCCR description.

Table 134: **Absolute Timer Cycle Count Register Bit Positions**

0-19	20-25	26-31
RESERVED	ATCMASK(5:0)	ATCID(5:0)

Table 135: **Absolute Timer Cycle Count Register Description**

Bits	Name	Access	Default Value	Description
0-19	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
20-25	ATCMASK(5:0)	Read/Write	0	**Absolute Timer Cycle Mask**: Indicates the Cycle ID Mask for the Absolute timer. These bits can be written to only when ATEN bit in the ATCR is '0.'
26-31	ATCID(5:0)	Read/Write	0	**Absolute Timer Cycle ID**: Indicates the Cycle ID field for the Absolute timer. These bits can be written to only when ATEN bit in the ATCR is '0'.

Absolute Timer Macrotick Offset Register (ATMOR)

The Absolute Timer Macrotick Offset Register (ATMOR) indicates the Macrotick offset and the operational mode for the Absolute timer.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in the Normal Active or Normal Passive operational state only when the ATEN bit in the ATCR is a '0.

Table 136 shows the bit positions in the ATMOR and Table 137 provides the ATMOR description.

Table 136: **Absolute Timer Macrotick Offset Register Bit Positions**

0-16	17	18-31
RESERVED	ATMODE	ATMAOFF(13:0)

Table 137: **Absolute Timer Macrotick Offset Register Description**

Bits	Name	Access	Default Value	Description
0-16	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
17	ATMODE	Read/Write	0	**Absolute Timer Mode**: This bit defines the operational mode of the Absolute timer. '1' = Repetitive. In the repetitive mode, after the timer elapses, the timer restarts. '0' = Non-Repetitive. In the non-repetitive mode, after the timer elapses, the timer does not restart.
18-31	ATMAOFF(13:0)	Read/Write	0	**Absolute Timer Offset**: Indicates the Macrotick Offset.

Relative Timer Control Register (RTCR)

The Relative Timer Control Register (RTCR) contains the enable bit for the Relative timer. The enable bit is cleared to a '0' by the controller if the operational mode of the timer is non-repetitive. Writing a '0' to the enable bit disables the Relative timer.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in Normal Active or Normal Passive operational states.

Table 138 shows the bit positions in the RTCR and Table 139 provides the RTCR description.

Table 138: **Relative Timer Control Register Bit Positions**

0-30	31
RESERVED	RTEN

Table 139: **Relative Timer Control Register Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
31	RTEN	Read/Write		**Relative Timer Enable**: The RTEN bit acts as an enable for the Relative Timer. '1' = Writing a '1' to this bit enables the Relative timer. '0' = Writing a '0' to this bit disables the Relative timer. Reads to this bit, indicate whether the Relative timer is enabled or disabled. The RTEN bit is cleared to a '0' by the FlexRay controller after the timer elapses only when the RTMODE bit in the RTMOR is a '0.'

Relative Timer Macrotick Offset Register (RTMOR)

The Relative Timer Macrotick Offset Register (RTMOR) indicates the Macrotick offset and the operational mode for the Relative timer. The Macrotick offset bit field should never be configured to all '0s.'

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in the Normal Active or Normal Passive operational states only when the RTEN bit in the RTCR is a '0'.

Table 140 shows the bit positions in the RTMOR and Table 141 provides the RTMOR description.

Table 140: **Relative Timer Macrotick Offset Register Bit Positions**

0	1-31
RTMODE	RTMAOFF(30:0)

Table 141: **Relative Timer Macrotick Offset Register Description**

Bits	Name	Access	Default Value	Description
0	RTMODE	Read/Write	0	**Relative Timer Mode**: This bit defines the operational mode of the Relative timer. '1' = Repetitive. In the repetitive mode, after the timer elapses, the timer restarts. '0' = Non-Repetitive. In the non-repetitive mode, after the timer elapses, the timer does not restart.
1-31	RTMAOFF(30:0)	Read/Write	0X7FFFFFFF	**Relative Timer Offset**: Indicates the Macrotick Offset. Writing a value of 0X00000000 to this bit field is prohibited.

Controller Status Registers

Aggregated Status Register (ASR)

The Aggregated Status Register (ASR) provides the host with an accrued status of channel activity for all the slots, regardless of whether the associated message is stored.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed. The bits in the ASR are cleared by the controller after a read operation.

Table 142 shows the bit positions in the ASR and Table 143 provides the ASR description.

Table 142: **Aggregated Status Register Bit Positions**

0-26	27	28	29	30	31
RESERVED	ASYNER	ACONER	AVALFR	AFRMER	ABDVIOL

Table 143: **Aggregated Status Register Description**

Bits	Name	Access	Default Value	Description
0-26	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
27	ASYNER	Read	0	**Aggregate Syntax Error**: Indicates aggregated syntax error status '1' = A syntax error has been observed since the last read to this register. '0' = No syntax errors have been observed since the last read to this register. This bit is cleared following a read operation.
28	ACONER	Read	0	**Aggregate Content Error**: Indicates aggregated content error status '1' = A content error has been observed since the last read to this register. '0' = No content errors have been observed since the last read to this register. This bit is cleared following a read operation.

Bits	Name	Access	Default Value	Description
29	AVALFR	Read	0	**Aggregate Valid Frame**: Indicates aggregated valid frame status '1' = One or more valid frames have been decoded since the last read to this register. '0' = No valid frames have been decoded since the last read to this register. This bit is cleared following a read operation.
30	AFRMER	Read	0	**Aggregate Frame Error**: Indicates the additional communication Indicator bit. '1' = A valid frame has been decoded in a slot that had encountered one or more of the following errors: ² boundary violation ² syntax error ² content error '0' = Valid frame has not been decoded in a slot that had additional channel activity. This bit is cleared following a read operation.
31	ABDVIOL	Read	0	**Aggregate Boundary Violation**: Indicates boundary violation status '1' = A boundary violation has been observed since the last read to this register. '0' = No boundary violations have been observed since the last read to this register. This bit is cleared following a read operation.

NIT Status Register (NSR)

The NSR indicates the occurrence of syntax errors and boundary violations during the NIT.

Access Criterion for Reads/Writes

- Read—Allowed in all operational states.
- Writes—Not allowed.

Table 144 Shows the bit positions in the NSR and Table 145 provides the NSR description.

Table 144: **NIT Status Register Bit Positions**

0-29	30	31
RESERVED	NSYNER	NBDVIOL

Table 145: **NIT Status Register Description**

Bits	Name	Access	Default Value	Description
0-29	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
30	NSYNER	Read	0	**NIT Syntax Error**: Indicate the occurrence of a syntax error during NIT. '1' = Indicates that a syntax error has been observed during the NIT since the last read to this register. '0' = Indicates that no syntax errors have been observed during the NIT since the last read to this register. This bit is cleared following a read operation.
31	NBDVIOL	Read		**NIT Boundary Violation**: Indicate the occurrence of a boundary violation during NIT. '1' = Indicates that a boundary violation has been observed during the NIT since the last read to this register. '0' = Indicates that no boundary violations have been observed during the NIT since the last read to this register. This bit is cleared following a read operation.

Symbol Window Status Register (SWSR)

The SWSR indicates the occurrence of valid symbols, syntax errors, and boundary violations during Symbol Window.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 146 shows the bit positions in the SWSR and Table 147 provides the SWSR description.

Table 146: **Symbol Window Status Register Bit Positions**

0-27	28	29	30	31
RESERVED	SVALSYM	SSYNER	SBDVIOL	STXCON

Table 147: **Symbol Window Status Register Description**

Bits	Name	Access	Default Value	Description
0-27	RESERVED	Read/ Write	0	**Reserved**: These bit positions are reserved for future expansion.
28	SVALSYM	Read	0	**Symbol Window Valid Symbol**: Indicate the occurrence of a valid symbol during Symbol Window. '1' = Indicates that a valid symbol has been decoded during the Symbol Window since the last read to this register. '0' = Indicates that no symbols have been observed during the Symbol Window since the last read to this register. This bit is cleared following a read operation.

Bits	Name	Access	Default Value	Description
29	SSYNER	Read	0	**Symbol Window Syntax Error**: Indicates the occurrence of a syntax error during Symbol Window. '1' = Indicates that a syntax error has been observed during the Symbol Window since the last read to this register. '0' = Indicates that no syntax errors have been observed during the Symbol Window since the last read to this register. This bit is cleared following a read operation.
30	SBDVIOL	Read	0	**Symbol Window Boundary Violation:** Indicates the occurrence of a boundary violation during Symbol Window. '1' = Indicates that a boundary violation has been observed during the Symbol Window since the last read to this register. '0' = Indicates that no boundary violations have been observed during the Symbol Window since the last read to this register. This bit is cleared following a read operation.
31	STXCON	Read	0	**Symbol Window Tx Conflict:** '1' = Indicates that a Tx Conflict has been observed during the Symbol Window since the last read to this register. '0' = Indicates that no Tx Conflicts have been observed during the Symbol Window since the last read to this register. This bit is cleared following a read operation.

Network Management Vector Registers[1..3] (NMVR[n])

The Network Management Vector Registers (NMVR[n]) store the accumulated bitwise OR'd received network management vectors. The length of the Network Management Vector is indicated by the NMVLR. In case the value indicated by the NMVLR is less than 12 bytes, the rest of the bytes are '0s.' NMVR are updated at the end of every communication cycle with the accrued NMV received in the current communication cycle, while the core is in Normal Active or Normal Passive states.

THe NMVR stores bytes as follows:

- NMVR[1] stores bytes 1, 2, 3 and 4 of the received Network Management Vector.
- NMVR[2] stores bytes 5, 6, 7 and 8 of the received Network Management Vector.
- NMVR[3] stores bytes 9, 10, 11 and 12 of the received Network Management Vector.

The address of NMVR[n] can be computed as:

Address of NMVR[n] = 0x"00000470" + (n-1)*0x"00000004"

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 148 shows the bit positions in the NMVR[n] and Table 149 provides the NMVR[n] description.

Table 148: **Network Management Vector Register[n] Bit Position**

0-7	8-15	16-23	24-31
NMVB1	NMVB2	NMVB3	NMVB4

Table 149: **Network Management Vector Register[n] Description**

Bits	Name	Access	Default Value	Description
0-7	NMVB1	Read	0	**Network Management Vector Byte 1**: Indicates the received Network Management Vector. In NMVR[1], indicates byte 1 of the received Network Management Vector. In NMVR[2], indicates byte 5 of the received Network Management Vector. In NMVR[3], indicates byte 9 of the received Network Management Vector.

Bits	Name	Access	Default Value	Description
8-15	NMVB2	Read	0	**Network Management Vector Byte 2**: Indicates the received Network Management Vector. In NMVR[1], indicates byte 2 of the received Network Management Vector. In NMVR[2], indicates byte 6 of the received Network Management Vector. In NMVR[3], indicates byte 10 of the received Network Management Vector.
16-23	NMVB3	Read	0	**Network Management Vector Byte 3**: Indicates the received Network Management Vector. In NMVR[1], indicates byte 3 of the received Network Management Vector. In NMVR[2], indicates byte 7 of the received Network Management Vector. In NMVR[3], indicates byte 11 of the received Network Management Vector.
24-31	NMVB4	Read	0	**Network Management Vector Byte 4**: Indicates the received Network Management Vector. In NMVR[1], indicates byte 4 of the received Network Management Vector. In NMVR[2], indicates byte 8 of the received Network Management Vector. In NMVR[3], indicates byte 12 of the received Network Management Vector.

Receive Slot Status Registers[1..512] (RSSR[n])

The Receive Slot Status Registers[1..512] (`RSSR[n]`) provides slot related status information such as errors, null frames for all the configured slots. A single register RSSR[n] provides the slot status information for slots 4n, 4n-1, 4n-2, 4n-3. 'N' can take values from 1 to 512. For example, `RSSR[1]` stores the slot status information for slots 4, 3, 2, 1, and `RSSR[2]` stores the slot status information for slots 8, 7, 6, 5.

The address of the Slot Status Register for a slot number 'n' which is a multiple of four can be computed as follows:

RSSR[1] = `0x00000500`

RSSR[n /4] = `0x00000500` + ((n /4) -1) x 4

Since each Slot Status Register contains slot status information for two even-odd slot pair, reading `RSSR[n]` would give the status information for slots 4n, 4n-1; 4n-2 and 4n-3.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.

- Writes—Not allowed.

Table 150 , Table 151, Table 152 and Table 153 show the bit positions in the RSSR[n] and Table 154 provides the RSSR[n] description.

Table 150: **Receive Slot Status Register[n] Bit Positions**

0	1	2	3	4	5	6	7
RESERVED	RSSTXCON	RSSPPIND	RSSNFI	RSSSYNER	RSSCONER	RSSBDVIOL	RSSVALFR

Table 151: **Receive Slot Status Register[n] Bit Positions**

8	9	10	11	12	13	14	15
RESERVED	RSSTXCON	RSSPPIND	RSSNFI	RSSSYNER	RSSCONER	RSSBDVIOL	RSSVALFR

Table 152: **Receive Slot Status Register[n] Bit Positions**

16	17	18	19	20	21	22	23
RESERVED	RSSTXCON	RSSPPIND	RSSNFI	RSSSYNER	RSSCONER	RSSBDVIOL	RSSVALFR

Table 153: **Receive Slot Status Register[n] Bit Positions**

24	25	26	27	28	29	30	31
RESERVED	RSSTXCON	RSSPPIND	RSSNFI	RSSSYNER	RSSCONER	RSSBDVIOL	RSSVALFR

Table 154: **Receive Slot Status Register[n] Description**

Bits	Name	Access	Default Value	Description
0	RESERVED	Read	0	**Reserved**: This bit position is reserved for future expansion.
1	RSSTXCON	Read	0	**Receive Slot Status Tx Conflict**: Indicates Tx Conflict Error for slot 4n-3. '1' = Indicates that a Tx Conflict was observed during the slot 4n-3. "0' = Indicates that Tx Conflicts were not observed during the slot 4n-3.
2	RSSPPIND	Read	0	**Receive Slot Status Payload Preamble Indicator**: Indicates the Payload Preamble indicator bit that was received for slot 4n-3.

Bits	Name	Access	Default Value	Description
3	RSSNFI	Read	0	**Receive Slot Status Null Frame Indicator**: Indicates the Null Frame indicator bit that was received for slot 4n-3.
4	RSSSYNER	Read	0	**Receive Slot Status Syntax Error**: Indicates the occurrence of a Syntax Error during slot 4n-3. '1' = Indicates that a Syntax Error was observed during the slot 4n-3. "0' = Indicates that a Syntax Error was not observed during the slot 4n-3.
5	RSSCONER	Read	0	**Receive Slot Status Content Error**: Indicates the occurrence of a Content Error during slot 4n-3. '1' = Indicates that a Content Error was observed during the slot 4n-3. "0' = Indicates that a Content Error was not observed during the slot 4n-3.
6	RSSBDVIOL	Read	0	**Receive Slot Status Boundary Violation**: Indicates the occurrence of a Boundary violation during slot 4n-3 '1' = Indicates that a Content Error was observed during the slot 4n-3. "0' = Indicates that a Content Error was not observed during the slot 4n-3.
7	RSSVALFR	Read	0	**Receive Slot Status Valid Frame**: Indicates the reception of a valid frame during slot 4n-3. '1' = Indicates that a valid frame was received during slot 4n-3. "0'= Indicates that a valid frame was not received during slot 4n-3.
8	RESERVED	Read	0	**Reserved**: This bit position is reserved for future expansion.

Bits	Name	Access	Default Value	Description
9	RSSTXCON	Read	0	**Receive Slot Status Tx Conflict**: Indicates Tx Conflict Error for slot 4n-2 '1' = Indicates that a Tx Conflict was observed during the slot 4n-2. "0'= Indicates that Tx Conflicts were not observed during the slot 4n-2.
10	RSSPPIND	Read	0	**Receive Slot Status Payload Preamble Indicator**: Indicates the Payload Preamble indicator bit that was received for slot 4n-2.
11	RSSNFI	Read	0	**Receive Slot Status Null Frame Indicator**: Indicates the Null Frame indicator bit that was received for slot 4n-2.
12	RSSSYNER	Read	0	**Receive Slot Status Syntax Error**: Indicates the occurrence of a Syntax Error during slot 4n-2 '1' = Indicates that a Syntax Error was observed during the slot 4n-2. "0'= Indicates that a Syntax Error was not observed during the slot 4n-2.
13	RSSCONER	Read	0	**Receive Slot Status Content Error**: Indicates the occurrence of a Content Error during slot 4n-2 '1' = Indicates that a Content Error was observed during the slot 4n-2. "0'= Indicates that a Content Error was not observed during the slot 4n-2.
14	RSSBDVIOL	Read	0	**Receive Slot Status Boundary Violation**: Indicates the occurrence of a Boundary violation during 4n-2 '1' = Indicates that a Content Error was observed during the slot 4n-2. "0'= Indicates that a Content Error was not observed during the slot 4n-2.

Bits	Name	Access	Default Value	Description
15	RSSVALFR	Read	0	**Receive Slot Status Valid Frame**: Indicates the reception of a valid frame during slot 4n-2 '1' = Indicates that a valid frame was received during slot 4n-2. "0'= Indicates that a valid frame was not received during slot 4n-2.
16	RESERVED	Read/Write	0	**Reserved**: This bit position is reserved for future expansion.
17	RSSTXCON	Read	0	**Receive Slot Status Tx Conflict**: Indicates Tx Conflict Error for slot 4n-1. '1' = Indicates that a Tx Conflict was observed during the slot 4n-1. "0'= Indicates that Tx Conflicts were not observed during the slot 4n-1.
18	RSSPPIND	Read	0	**Receive Slot Status Payload Preamble Indicator**: Indicates the Payload Preamble indicator bit that was received for slot 4n-1.
19	RSSNFI	Read	0	**Receive Slot Status Null Frame Indicator**: Indicates the Null Frame indicator bit that was received for slot 4n-1.
20	RSSSYNER	Read	0	**Receive Slot Status Syntax Error:** Indicates the occurrence of a Syntax Error during slot 4n-1. '1' = Indicates that a Syntax Error was observed during the slot 4n-1. "0'= Indicates that a Syntax Error was not observed during the slot 4n-1.
21	RSSCONER	Read	0	**Receive Slot Status Content Error**: Indicates the occurrence of a Content Error during slot 4n-1. '1' = Indicates that a Content Error was observed during the slot 4n-1. "0'= Indicates that a Content Error was not observed during the slot 4n-1.

Bits	Name	Access	Default Value	Description
22	RSSBDVIOL	Read	0	**Receive Slot Status Boundary Violation:** Indicates the occurrence of a Boundary violation during slot 4n-1 '1' = Indicates that a Content Error was observed during the slot 4n-1. "0'= Indicates that a Content Error was not observed during the slot 4n-1.
23	RSSVALFR	Read	0	**Receive Slot Status Valid Frame**: Indicates the reception of a valid frame during slot 4n-1. '1' = Indicates that a valid frame was received during slot 4n-1. "0'= Indicates that a valid frame was not received during slot 4n-1.
24	RESERVED	Read	0	**Reserved**: This bit position is reserved for future expansion.
25	RSSTXCON	Read	0	**Receive Slot Status Tx Conflict**: Indicates Tx Conflict Error for slot 4n '1' = Indicates that a Tx Conflict was observed during the slot 4n. "0'= Indicates that Tx Conflicts were not observed during the slot 4n.
26	RSSPPIND	Read	0	**Receive Slot Status Payload Preamble Indicator**: Indicates the Payload Preamble indicator bit that was received for slot 4n.
27	RSSNFI	Read	0	**Receive Slot Status Null Frame Indicator**: Indicates the Null Frame indicator bit that was received for slot 4n.
28	RSSSYNER	Read	0	**Receive Slot Status Syntax Error**: Indicates the occurrence of a Syntax Error during slot 4n '1' = Indicates that a Syntax Error was observed during the slot 4n. "0'= Indicates that a Syntax Error was not observed during the slot 4n.

Bits	Name	Access	Default Value	Description
29	RSSCONER	Read	0	**Receive Slot Status Content Error:** Indicates the occurrence of a Content Error during slot 4n '1' = Indicates that a Content Error was observed during the slot 4n. "0'= Indicates that a Content Error was not observed during the slot 4n.
30	RSSBDVIOL	Read	0	**Receive Slot Status Boundary Violation:** Indicates the occurrence of a Boundary violation during slot 4n. '1' = Indicates that a Content Error was observed during the slot 4n. "0'= Indicates that a Content Error was not observed during the slot 4n.
31	RSSVALFR	Read	0	**Receive Slot Status Valid Frame:** Indicates the reception of a valid frame during slot 4n. '1' = Indicates that a valid frame was received during slot 4n. "0'= Indicates that a valid frame was not received during slot 4n.

Interrupt Control and Status Registers

FlexRay Interrupt Control and Status Registers

The FlexRay controller uses a hard-vector interrupt mechanism with four interrupt lines (`Flex_Cirpt`, `Flex_Txbirpt`, `Flex_Rxbirpt`, `Flex_Rxfirpt`) to indicate interrupts. Interrupts are indicated by asserting the interrupt lines (transition of the interrupt lines from a logic '0' to a logic '1').

Events such as errors on the bus line, message transmission and reception, and FIFO overflow and underflow conditions can generate interrupts. During power on, the Interrupt lines are driven low. Each of these lines is associated with an Interrupt Status Register, Interrupt Enable Register, and Interrupt Clear Register.

General Interrupt Description

The Interrupt Status Registers (ISRs) indicate the interrupt status bits. These bits are set and cleared regardless of the status of the corresponding bit in the Interrupt Enable Registers (IERs). The IERs handle the interrupt-enable functionality. Status bits in the ISRs are cleared by writing a '1' to the corresponding bits in the Interrupt Clear Registers (ICRs). Setting an ICR bit clears the interrupt line regardless of the corresponding ISR when the IER is set. The interrupt line maybe reasserted if there are pending uncleared ISR bits. This applies to all the interrupt lines.

Triggering Interrupts

The following conditions cause the interrupt lines to be asserted:

- If a bit in the ISR is '1,' and the corresponding bit in the IER is '1.'
- Changing an IER bit from a '0' to '1' when the corresponding bit in the ISR is currently '1.'
- Setting an ICR bit whose corresponding ISR bit is high.

Clearing Interrupts

Two conditions cause the interrupt lines to be deasserted:

- Clearing a bit in the ISR that is '1' (by writing a '1' to the corresponding bit in the ICR) when the corresponding bit in the IER is '1'
- Changing an IER bit from '1' to '0' when the corresponding bit in the ISR is '1'.

When both deassertion and assertion conditions occur simultaneously, the interrupt lines are deasserted first, and reasserted if the assertion condition remains true.

Individual Interrupt Control and Status Registers

The following registers are associated with each of the Interrupt Lines:

- The `CISR`, `CIER`, `CICR` define the Interrupt Status bits, Interrupt Enable bits, and Interrupt Clear bits respectively for the `Flex_Cirpt` interrupt line.
- The `TXBISR[n]`, `TXBIER[n]`, `TXBICR[n]` define the Interrupt Status bits, Interrupt Enable bits and Interrupt Clear bits respectively for the `Flex_Txbirpt` interrupt line.
- The `RXBISR[n]`, `RXBIER[n]`, `RXBICR[n]` define the Interrupt Status bits, Interrupt Enable bits and Interrupt Clear bits respectively for the `Flex_Txbirpt` interrupt line.
- The `RXFISR`, `RXFIER`, `RXFICR` define the Interrupt Status bits, Interrupt Enable bits and Interrupt Clear bits respectively for the `Flex_Rxfirpt` interrupt line.

Controller Interrupt Status Register (CISR)

The Controller Interrupt Status Register (CISR) indicates the following interrupt conditions:

- A Fatal Protocol Error has occurred
- A Protocol Error has occurred
- Relative and Absolute Timers have elapsed
- Communication Cycle has ended
- Clock correction failures
- Changes in the operational state of the controller
- Changes in Error mode of the controller
- Reception of Sync frames
- Reception of WUP pattern
- Reception of MTS
- Cold start abortion

The bits in the CISR are set irrespective of the status of the corresponding bits in the CIEN. Writing to the CICR clears the bits that are set to a '1' in the CISR.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.

- Writes—Not allowed.

Table 155 shows the bit positions in the CISR and Table 156 provides the CISR description.

Table 155: **Controller Interrupt Status Register Bit Positions**

0-16	17	18	19	20	21	22	23
RESERVED	RELTIM	ABTIM	PLTXVIOL	BVIOL	PERR	SYNCOFLW	EXBOUND

24	25	26	27	28	29	30	31
MITERM	CSABORT	WUPRX	MTSRX	SYNCRX	POCCHNG	EMCHNG	CSSTART

Table 156: **Controller Interrupt Status Register Description**

Bits	Name	Access	Default Value	Description
0-16	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
17	RELTIM	Read	0	**Relative Timer Interrupt**: This bit indicates the elapse of the Relative timer. '1' = Relative timer has elapsed. '0' = Relative timer has not elapsed since the last clear operation on this bit.
18	ABTIM	Read	0	**Absolute Timer Interrupt**: This bit indicates the elapse of the Absolute timer. '1' = Absolute timer has elapsed. '0' = Absolute timer has not elapsed since the last clear operation on this bit.
19	PLTXVIOL	Read	0	**Latest Tx Violation Interrupt:** This bit corresponds to the protocol status bit pLatestTx violation. '1' = Indicates a Fatal Protocol Error due to the pLatestTx violation condition. '0' = pLatestTx violation has not been set since the last clear operation on this bit.

Bits	Name	Access	Default Value	Description
20	BVIOL	Read	0	**Boundary Violation Interrupt**: This bit corresponds to the protocol status bit vSS!BViolation. '1' = Indicates a Fatal Protocol Error due to a boundary violation. '0' = A boundary violation resulting in a Fatal Protocol Error has not been observed since the last clear operation on this bit.
21	PERR	Read	0	**Protocol Error Interrupt**: This bit indicates a protocol error condition. '1'= Indicates a protocol error (Syntax error, Content error, Tx Conflict, Boundary Violation that does not result in a Fatal protocol error). '0' = Indicates that a protocol error was not observed since the last clear operation on this bit.
22	SYNCOFLW	Read	0	**Sync Frame Overflow Interrupt:** This bit indicates a sync frame overflow condition. '1' = Indicates that more than 15 sync frames have been received during a communication cycle. '0' = Indicates that more than 15 sync frames have not been received during a communication cycle since the last clear operation on this bit.

Bits	Name	Access	Default Value	Description
23	EXBOUND	Read	0	**EXCEED_BOUNDS Interrupt**: This bit indicates that the computed rate and offset correction values exceed the limits defined in the RCLR and OCLR registers respectively '1' = Computed Rate and Offset correction values exceed the limit. '0' = Computed Rate and Offset correction values have not exceeded the limit since the last clear operation on this bit.
24	MITERM	Read	0	**MISSING_TERM Interrupt**: This bit indicates that rate and offset correction values could not be computed. '1' = Rate and Offset correction terms could not be calculated. '0' = Rate and Offset correction terms could be calculated since the last clear operation on this bit.
25	CSABORT	Read	0	**Cold Start Abort Interrupt:** This bit corresponds to the protocol status parameter zColdStartAborted. '1' = Indicates that the cold start operation has been aborted. '0' = Indicates that the cold start operation has not been aborted since the last clear operation on this bit.
26	WUPRX	Read	0	**WUP Symbol Received Interrupt**: Indicates the reception of a wake up symbol. '1' = Indicates that a wake up symbol has been received. '0'= Indicates that a wake up symbol has not been received since the last clear operation on this bit.

Bits	Name	Access	Default Value	Description
27	MTSRX	Read	0	**MTS Symbol Received Interrupt**: Indicates the reception of a symbol during the symbol window. '1' = Indicates that a symbol has been received during the symbol window. '0'= Indicates that a symbol has not been received during the symbol window since the last clear operation on this bit.
28	SYNCRX	Read	0	**Sync Frame Received Interrupt:** Indicates the reception of a sync frame. '1' = Indicates that a sync frame has been received. '0'= Indicates that a sync frame has not been received since the last clear operation on this bit.

Bits	Name	Access	Default Value	Description
29	POCCHNG	Read	0	**POC State Change Interrupt:** Indicates a change in the protocol status parameter vPOC!State. '1' = Indicates that the vPOC!State parameter has changed. '0' = Indicates that the vPOC!State parameter has not changed since the last clear operation on this bit.
30	EMCHNG	Read	0	**Error Mode Change Interrupt:** Indicates a change in the protocol status parameter vPOC!ErrorMode. '1' = Indicates that the vPOC!ErrorMode parameter has changed. '0' = Indicates that the vPOC!Error-Mode parameter has not changed since the last clear operation on this bit.
31	CSSTART	Read	0	**Communication Cycle Start Interrupt:** Indicates a change in the protocol status parameter vCycleCounter. '1' = Indicates the start of a new communication cycle '0' = Indicates that the communica-tion cycle has not changed since the last clear operation on this bit.

Controller Interrupt Enable Register (CIER)

The Controller Interrupt Enable Register (CIER) contains bits that are used to enable interrupt generation. Interrupt events are indicated by the assertion of the `Flex_Cirpt` interrupt line.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Table 157 shows the bit positions in the CIER and Table 158 provides the CIER description.

Table 157: **Controller Interrupt Enable Register Bit Positions**

0-16	17	18	19	20	21	22	23
RESERVED	RELTIME	ABTIME	PLTXVIOLE	BVIOLE	PERRE	SYNCOFLWE	EXBOUNDE

24	25	26	27	28	29	30	31
MITERME	CSABORTE	WUPRXE	MTSRXE	SYNCRXE	POCCHNGE	EMCHNGE	CSSTARTE

Table 158: **Controller Interrupt Enable Register Bit Fields**

Bits	Name	Access	Default Value	Description
0-16	RESERVED	Read/Write	0	**Reserved:** These bit positions are reserved for future expansion
17	RELTIME	Read/Write	0	**Relative Timer Interrupt Enable**: Interrupt Enable bit for the RELTIM bit in the CISR. '1' = The Flex_Cirpt line is asserted when the RELTIM bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the RELTIM bit in the CISR is '1.'
18	ABTIME	Read/Write	0	**Absolute Timer Interrupt Enable:** Interrupt Enable bit for the ABTIM bit in the CISR. '1' = The Flex_Cirpt line is asserted when the ABTIM bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the ABTIM bit in the CISR is '1.'
19	PLTXVIOLE	Read/Write	0	**Latest Tx Violation Interrupt Enable:** Interrupt Enable bit for the PLTXVIOL bit in the CISR. '1' = The Flex_Cirpt line is asserted when the PLTXVIOL bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the PLTXVIOL bit in the CISR is '1.'
20	BVIOLE	Read/Write	0	**Boundary Violation Interrupt Enable:** Interrupt Enable bit for the BVIOL bit in the CISR. '1' = The Flex_Cirpt line is asserted when the BVIOL bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the BVIOL bit in the CISR is '1.'

Table 158: **Controller Interrupt Enable Register Bit Fields Œ[} œç**

Bits	Name	Access	Default Value	Description
21	PERRE	Read/Write	0	**Protocol Error Interrupt Enable**: Interrupt Enable bit for the PERR bit in the CISR. '1' = The Flex_Cirpt line is asserted when the PERR bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the PERR bit in the CISR is '1.'
22	SYNCOFLWE	Read/Write	0	**Sync Frame Overflow Interrupt Enable**: Interrupt Enable bit for the SYNCOFLW bit in the CISR. '1' = The Flex_Cirpt line is asserted when the SYNCOFLW bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the SYNCOFLW bit in the CISR is '1.'
23	EXBOUNDE	Read/Write	0	**EXCEED_BOUNDS Interrupt Enable:** Interrupt Enable bit for the EXBOUND bit in the CISR. '1' = The Flex_Cirpt line is asserted when the EXBOUND bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the EXBOUND bit in the CISR is '1.'
24	MITERME	Read/Write	0	**MISSING_TERM Interrupt Enable:** Interrupt Enable bit for the MITERM bit in the CISR. '1' = The Flex_Cirpt line is asserted when the MITERM bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the MITERM bit in the CISR is '1.'
25	CSABORTE	Read/Write	0	**Cold Start Abort Interrupt Enable:** Interrupt Enable bit for the CSABORT bit in the CISR. '1' = The Flex_Cirpt line is asserted when the CSABORT bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the CSABORT bit in the CISR is '1.'

Table 158: **Controller Interrupt Enable Register Bit Fields Œ[} œç**

Bits	Name	Access	Default Value	Description
26	WUPRXE	Read/Write	0	**WUP Symbol Received Interrupt Enable:** Interrupt Enable bit for the WUPRX bit in the CISR. '1' = The Flex_Cirpt line is asserted when the WUPRX bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the WUPRX bit in the CISR is '1.'
27	MTSRXE	Read/Write	0	**MTS Symbol Received Interrupt Enable:** Interrupt Enable bit for the MTSRX bit in the CISR. '1' = The Flex_Cirpt line is asserted when the MTSRX bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the MTSRX bit in the CISR is '1.'
28	SYNCRXE	Read/Write	0	**Sync Frame Received Interrupt Enable:** Interrupt Enable bit for the SYNCRX bit in the CISR. '1' = The Flex_Cirpt line is asserted when the SYNCRX bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the SYNCRX bit in the CISR is '1.'
29	POCCHNGE	Read/Write	0	**POC State Change Interrupt Enable:** Interrupt Enable bit for the POCCHNG bit in the CISR. '1' = The Flex_Cirpt line is asserted when the POCCHNG bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the POCCHNG bit in the CISR is '1'.

Table 158: **Controller Interrupt Enable Register Bit Fields Œ[} œç**

Bits	Name	Access	Default Value	Description
30	EMCHNGE	Read/Write	0	**Error Mode Change Interrupt Enable:** Interrupt Enable bit for the EMCHNG bit in the CISR. '1' = The Flex_Cirpt line is asserted when the EMCHNG bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the EMCHNG bit in the CISR is '1'.
31	CSSTARTE	Read/Write	0	**Communication Cycle Start Interrupt Enable:** Interrupt Enable bit for the CSSTART bit in the CISR. '1' = The Flex_Cirpt line is asserted when the CSSTART bit in the CISR is '1.' '0' = The Flex_Cirpt line is not asserted when the CSSTART bit in the CISR is '1.'

Controller Interrupt Clear Register (CICR)

The Controller Interrupt Clear Register (CICR) contains bits that are used to clear the bits in the CISR.

Access Criterion for Reads/Writes

- Reads—Not Allowed.
- Writes—Allowed in all operational states.

Table 159 shows the bit positions in the CICR and Table 160 provides the CICR description.

Table 159: **Controller Interrupt Clear Register Bit Positions**

0-16	17	18	19	20	21	22	23
RESERVED	RELTIMC	ABTIMC	PLTXVIOLC	BVIOLC	PERRC	SYNCOFLWC	EXBOUNDC

24	25	26	27	28	29	30	31
MITERMC	CSABORTC	WUPRXC	MTSRXC	SYNCRXC	POCCHNGC	EMCHNGC	CSSTARTC

Table 160: **Controller Interrupt Clear Register Description**

Bits	Name	Access	Default Value	Description
0-16	RESERVED	Read/Write	0	**Reserved:** These bit position are reserved for future expansion.
17	RELTIMC	Write	0	**Relative Timer Interrupt Clear**: This bit is the clear bit for the RELTIM bit in the CISR. '1' = Writing a '1' to the RELTIMC bit clears the RELTIM bit in the CISR.
18	ABTIMC	Write	0	**Absolute Timer Interrupt Clear:** This bit is the clear bit for the ABTIM bit in the CISR. '1' = Writing a '1' to the ABTIMC bit clears the ABTIM bit in the CISR.
19	PLTXVIOLC	Write	0	**Latest Tx Violation Interrupt Clear**: This bit is the clear bit for the PLTXVIOL bit in the CISR. '1' = Writing a '1' to the PLTXVIOLC bit clears the PLTXVIOL bit in the CISR.
20	BVIOLC	Write	0	**Boundary Violation Interrupt Clear**: This bit is the clear bit for the BVIOL bit in the CISR. '1' = Writing a '1' to the BVIOLC bit clears the BVIOL bit in the CISR.
21	PERRC	Write	0	**Protocol Error Interrupt Clear**: This bit is the clear bit for the PERR bit in the CISR. '1' = Writing a '1' to the PERRC bit clears the PERR bit in the CISR.
22	SYNCOFLWC	Write	0	**Sync Frame Overflow Interrupt Clear**: This bit is the clear bit for the SYNCOFLW bit in the CISR. '1' = Writing a '1' to the SYNCOF-LWC bit clears the SYNCOFLW bit in the CISR.

Bits	Name	Access	Default Value	Description
23	EXBOUNDC	Write	0	**EXCEED_BOUNDS Interrupt Clear:** This bit is the clear bit for the EXBOUND bit in the CISR. '1' = Writing a '1' to the EXBOUNDC bit clears the EXBOUND bit in the CISR.
24	MITERMC	Write	0	**MISSING_TERM Interrupt Clear**: This bit is the clear bit for the MITERM bit in the CISR. '1' = Writing a '1' to the MITERMC bit clears the MITERM bit in the CISR.
25	CSABORTC	Write	0	**Cold Start Abort Interrupt Clear:** This bit is the clear bit for the CSABORT bit in the CISR. '1' = Writing a '1' to the CSABORTC bit clears the CSABORT bit in the CISR.
26	WUPRXC	Write	0	**WUP Symbol Received Interrupt Clear:** This bit is the clear bit for the WUPRX bit in the CISR. '1' = Writing a '1' to the WUPRXC bit clears the WUPRX bit in the CISR.
27	MTSRXC	Write	0	**MTS Symbol Received Interrupt Clear:** This bit is the clear bit for the MTSRX bit in the CISR. '1' = Writing a '1' to the MTSRXC bit clears the MTSRX bit in the CISR.
28	SYNCRXC	Write	0	**Sync Frame Received Interrupt Clear**: This bit is the clear bit for the SYNCRX bit in the CISR. '1' = Writing a '1' to the SYNCRXC bit clears the SYNCRX bit in the CISR.

Bits	Name	Access	Default Value	Description
29	POCCHNGC	Write	0	**POC State Change Interrupt Clear:** This bit is the clear bit for the POCCHNG bit in the CISR. '1' = Writing a '1' to the POCCH-NGC bit clears the POCCHNG bit in the CISR.
30	EMCHNGC	Write	0	**Error Mode Change Interrupt Clear:** This bit is the clear bit for the EMCHNG bit in the CISR. '1' = Writing a '1' to the EMCHNGC bit clears the EMCHNG bit in the CISR.
31	CSSTARTC	Write	0	**Communication Cycle Start Interrupt Clear**: This bit is the clear bit for the CSSTART bit in the CISR. '1' = Writing a '1' to the CSSTARTC bit clears the CSSTART bit in the CISR.

Transmit Buffer Interrupt Status Register[1..4] (TXBISR[n])

The Transmit Buffer Interrupt Status Registers indicate the transmission status for the Transmit buffers. The number of Transmit Buffer Interrupt Status Registers and the number of bits in each of these registers depends on the parameter "Number of Tx Buffers". It should be mentioned that each Tx Buffer is assigned a particular bit in the TXBISR[n].

The address of Transmit Buffer Interrupt Status Registers can be computed as follows

Address of TXBISR[n] = `0x0000F10` + [n-1]*`0x00000004`

Access Criterion for Reads/Writes:

- **Reads**: Allowed in all the operational states.
- **Writes**: Not allowed

Table 161 provides the TXBISR[1] description. Table 162 provides the TXBISR[2] description.

Table 163 provides the TXBISR[3] description. Table 164 provides the TXBISR[4] description.

Table 161: **Transmit Buffer Interrupt Status Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	TXOK[31..0]	Read	0x0000	**Transmit OK Interrupt[31..0]**: TXOK[0: Corresponds to Transmit Buffer 1; TXOK[31] corresponds to Transmit Buffer 32. Each bit indicates that the message pended for transmission in the Transmit buffer has been successfully transmitted and that the Transmit buffer is now available for host updates. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that the message pended for transmission has been successfully transmitted. '0' = The message pended for transmission has not been transmitted since the last time this bit was cleared by writing to the TXBICR. Register Bits TXBISR[1][0..29] are invalid if Number of Tx Buffers is 2 Register Bits TXBISR[1][0..27] are invalid if Number of Tx Buffers is 4. Register Bits TXBISR[1][0..23] are invalid if Number of Tx Buffers is 8 Register Bits TXBISR[1][0..15] are invalid if Number of Tx Buffers is 16. All the register bits are valid if Number of Tx Buffers are greater than or equal to 32.

Table 162: **Transmit Buffer Interrupt Status Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOK[63..32]	Read	0x0000	**Transmit OK Interrupt[63..32]**: TXOK[32] corresponds to Transmit Buffer 33 while TXOK[63] corresponds to Transmit Buffer 64. Each bit indicates that the message pended for transmission in the Transmit buffer has been successfully transmitted and that the Transmit buffer is now available for host updates. There is a one to one correspondence between the bit position and the buffer number. '1' = Indicates that the message pended for transmission has been successfully transmitted. '0' = The message pended for transmission has not been transmitted since the last time this bit was cleared by writing to the TXBICR. This register is valid only when Number of Tx Buffers is greater than 32.

Table 163: **Transmit Buffer Interrupt Status Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	TXOK[95..64]	Read	0x0000	**Transmit OK Interrupt[64..95]**: TXOK[64] corresponds to Transmit Buffer 65 while TXOK[95] corresponds to Transmit Buffer 96. Each bit indicates that the message pended for transmission in the Transmit buffer has been successfully transmitted and that the Transmit buffer is now available for host updates. There is a one to one correspondence between the bit position and the buffer number. '1' = Indicates that the message pended for transmission has been successfully transmitted. '0' = The message pended for transmission has not been transmitted since the last time this bit was cleared by writing to the TXBICR. This register is valid only when Number of Tx Buffers is greater than 64.

Table 164: **Transmit Buffer Interrupt Status Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOK[127..96]	Read	0x0000	**Transmit OK Interrupt[127..96]**: TXOK[96] corresponds to Transmit Buffer 97 while TXOK[127] corresponds to Transmit Buffer 128. Each bit indicates that the message pended for transmission in the Transmit buffer has been successfully transmitted and that the Transmit buffer is now available for host updates. There is a one to one correspondence between the bit position and the buffer number. '1' = Indicates that the message pended for transmission has been successfully transmitted. '0' = The message pended for transmission has not been transmitted since the last time this bit was cleared by writing to the TXBICR. This register is valid only when Number of Tx Buffers is greater than 64.

Transmit Buffer Interrupt Enable Register[1..4] (TXBIER[n])

The Transmit Buffer Interrupt Enable Register (`TXBIER[n]`) indicates the interrupt enable bits for the `TXOK` bits in the `TXBISR[n]`. The number of Transmit Buffer Interrupt Enable Registers and the number of bits in each of these registers depends on the parameter `Number of Tx Buffers`. It should be mentioned that each `TXOK[x]` bit in the `TXBISR[n]` has a corresponding `TXOKE[x]` bit in the `TXBIER[n]`.

The address of Transmit Buffer Interrupt Enable Registers can be computed as follows:

Address of `TXBIER[n]` = `0x0000F20` + [n-1] × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Table 165 provides the TXBIER[1] description. Table 166 provides the TXBIER[2] description.

Table 167 provides the TXBIER[3] description. Table 168 provides the TXBIER[4] description.

Table 165: **Transmit Buffer Interrupt Enable Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	TXOKE[31..0]	Read/Write	0x0000	**Transmit OK Interrupt Enable [31..0]**: TXOKE[0] corresponds to TXOK[0] bit in the TXBISR[1] while TXOKE[31] corresponds to TXOK[31] bit in the TXBISR[1]. '1' = The Flex_Txbirpt line is asserted when the TXOK[x] bit in the TXBISR[1] is '1.' '0' = The Flex_Txbirpt line is not asserted when the TXOK[x] bit in the TXBISR[1] is '1.' Register Bits TXBIER[1][0..29] are invalid if Number of Tx Buffers is 2 Register Bits TXBIER[1][0..27] are invalid if Number of Tx Buffers is 4. Register Bits TXBIER[1][0..23] are invalid if Number of Tx Buffers is 8 Register Bits TXBIER[1][0..15] are invalid if Number of Tx Buffers is 16. All the register bits are valid if Number of Tx Buffers are greater than or equal to 32.

Table 166: **Transmit Buffer Interrupt Enable Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOKE[63..32]	Read/Write	0x0000	**Transmit OK Interrupt Enable [63..32]:** TXOKE[32] corresponds to TXOK[32] bit in the TXBISR[2] while TXOKE[63] corresponds to TXOK[63] bit in the TXBISR[2]. '1' = The Flex_Txbirpt line is asserted when the TXOK[x] bit in the TXBISR[2] is '1.' '0' = The Flex_Txbirpt line is not asserted when the TXOK[x] bit in the TXBISR[2] is '1.' This register is valid only when Number of Tx Buffers is greater than 32.

Table 167: **Transmit Buffer Interrupt Enable Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	TXOKE[95..64]	Read/Write	0x0000	**Transmit OK Interrupt Enable [95..64]**: TXOKE[64] corresponds to TXOK[64] bit in the TXBISR[3] while TXOKE[95] corresponds to TXOK[95] bit in the TXBISR[3]. '1' = The Flex_Txbirpt line is asserted when the TXOK[x] bit in the TXBISR[3] is '1.' '0' = The Flex_Txbirpt line is not asserted when the TXOK[x] bit in the TXBISR[3] is '1.' This register is valid only when Number of Tx Buffers is greater than 64.

Table 168: **Transmit Buffer Interrupt Enable Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOKE[127..96]	Read/Write	0x0000	**Transmit OK Interrupt Enable [127..96]**: TXOKE[96] corresponds to TXOK[96] bit in the TXBISR[4] while TXOKE[127] corresponds to TXOK[127] bit in the TXBISR[4]. '1' = The Flex_Txbirpt line is asserted when the TXOK[x] bit in the TXBISR[4] is '1.' '0'= The Flex_Txbirpt line is not asserted when the TXOK[x] bit in the TXBISR[4] is '1.' This register is valid only when Number of Tx Buffers is greater than 64.

Transmit Buffer Interrupt Clear Register[1..4] (TXBICR[n])

The Transmit Buffer Interrupt Clear Registers (`TXBICR[n]`) indicate the interrupt clear bits for the `TXOK` bits in the `TXBISR[n]`. The number of Transmit Buffer Interrupt Clear Registers and the number of bits in each of these registers depends on the value set for the parameter `Number of Tx Buffers`. Each `TXOK[x]` bit in the `TXBISR[n]` has a corresponding `TXOKC[x]` bit in the `TXBICR[n]`.

The address of Transmit Buffer Interrupt Clear Registers can be computed as follows:

Address of `TXBICR[n]` = `0x0000F30` + [n-1] × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Reads are not allowed.
- Writes—Allowed in all operational states.

Table 169 provides the TXBICR[1] description. Table 170 provides the TXBICR[2] description.

Table 171 provides the TXBICR[3] description. Table 172 provides the TXBICR[4] description.

Table 169: **Transmit Buffer Interrupt Clear Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	TXOKC[31..0]	Write	0x0000	**Transmit OK Interrupt Clear [31..0]**: TXOKC[0] corresponds to TXOK[0] bit in the TXBISR[1] while TXOKC[31] corresponds to TXOK[31] bit in the TXBISR[1]. '1' = Writing a '1' to the TXOKC[x] bit clears the TXOK[x] bit in the TXBISR[1]. Register Bits TXBICR[1][0..29] are invalid if Number of Tx Buffers is 2 Register Bits TXBICR[1][0..27] are invalid if Number of Tx Buffers is 4. Register Bits TXBICR[1][0..23] are invalid if Number of Tx Buffers is 8 Register Bits TXBICR[1][0..15] are invalid if Number of Tx Buffers is 16. All the register bits are valid if Number of Tx Buffers are greater than or equal to 32.

Table 170: **Transmit Buffer Interrupt Clear Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOKC[63..32]	Write	0x0000	**Transmit OK Interrupt Clear [63..32]**: TXOKC[32] corresponds to TXOK[32] bit in the TXBISR[2] while TXOKC[63] corresponds to TXOK[63] bit in the TXBISR[2]. '1' = Writing a '1' to the TXOKC[x] bit clears the TXOK[x] bit in the TXBISR[2]. This register is valid only when Number of Tx Buffers is greater than 32.

Table 171: **Transmit Buffer Interrupt Clear Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	TXOKC[95..64]	Write	0x0000	**Transmit OK Interrupt Clear [95..64]**: TXOKC[64] corresponds to TXOK[64] bit in the TXBISR[3] while TXOKC[95] corresponds to TXOK[95] bit in the TXBISR[3]. '1' = Writing a '1' to the TXOKC[x] bit clears the TXOK[x] bit in the TXBISR[3]. This register is valid only when Number of Tx Buffers is greater than 64.

Table 172: **Transmit Buffer Interrupt Clear Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	TXOKC[127..96]	Write	0x0000	**Transmit OK Interrupt Clear [127..96]**: TXOKC[96] corresponds to TXOK[96] bit in the TXBISR[4] while TXOKC[127] corresponds to TXOK[127] bit in the TXBISR[4]. '1' = Writing a '1' to the TXOKC[x] bit clears the TXOK[x] bit in the TXBISR[4]. This register is valid only when Number of Tx Buffers is greater than 64.

Receive Buffer Interrupt Status Register[1..8] (RXBISR[n])

The Receive Buffer Interrupt Status Registers (`RXBISR[n]`) indicate the message reception status for the Receive buffers. The number of Receive Buffer Interrupt Status Registers and the number of bits in each of these registers depends on the value set in the parameter `Number of Rx Buffers`. Each Rx Buffer is assigned a particular bit in the `RXBISR[n]`. `RXOK` bits indicate that a valid message has been written to the corresponding buffer. `RXOFW` bits indicate that the contents of the buffer with an uncleared `RXOK` bit have been overwritten with new data.

Note: If a receive buffer overflow event occurs on an already set RXBISR, the RXBISR will be temporarily taken low while the buffer contents are being updated with the latest received frame contents. The RXBISR will then be set again along with the RXOFLW bit. When RXBISR and RXOLFW bits have both both previously set and a new message passes the buffers filters, RXBISR behaves as described above, but RXOFLW bit remains asserted throughout.

The address of Receive Buffer Interrupt Status Registers can be computed as follows:

Address of `RXBISR[n]` = `0x0000F40` + [n-1] × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 173 provides the RXBISR[1] description. Table 174 provides the RXBISR[2] description.

Table 175 provides the RXBISR[3] description. Table 176 provides the RXBISR[4] description.

Table 177 provides the RXBISR[5] description. Table 178 provides the RXBISR[6] description.

Table 179 provides the RXBISR7] description. Table 180 provides the RXBISR[8] description.

Table 173: **Receive Buffer Interrupt Status Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOK[31..0]	Read	0x0000	**Receive OK Interrupt[31..0]**: RXOK[0] corresponds to Receive Buffer 1 while RXOK[31] corresponds to Receive Buffer 32. Each bit indicates that the a valid message is present in the Receive Buffer. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer. '0' = Indicates that a new message has not been received in the Receive buffer since the last clear operation. Register Bits RXBISR[1][0..29] are invalid if "Number of Rx Buffers" is 2. Register Bits RXBISR[1][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBISR[1][0..23] are invalid if "Number of Rx Buffers" is 8 Register Bits RXBISR[1][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 174: **Receive Buffer Interrupt Status Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOK[63..32]	Read	0x0000	**Receive OK Interrupt[63..32]**: RXOK[32] corresponds to Receive Buffer 33 while RXOK[63] corresponds to Receive Buffer 64. Each bit indicates that the a valid message is present in the Receive Buffer. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer. '0' = Indicates that a new message has not been received in the Receive buffer since the last clear operation. This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 175: **Receive Buffer Interrupt Status Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOK[95..64]	Read	0x0000	**Receive OK Interrupt[95..64]**: RXOK[64] corresponds to Receive Buffer 65 while RXOK[95] corresponds to Receive Buffer 96. Each bit indicates that the a valid message is present in the Receive Buffer. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer. '0' = Indicates that a new message has not been received in the Receive buffer since the last clear operation. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 176: **Receive Buffer Interrupt Status Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOK[127..96]	Read	0x0000	**Receive OK Interrupt[127..96]**: RXOK[96] corresponds to Receive Buffer 97 while RXOK[127] corresponds to Receive Buffer 128. Each bit indicates that the a valid message is present in the Receive Buffer. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer. '0' = Indicates that a new message has not been received in the Receive buffer since the last clear operation. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 177: **Receive Buffer Interrupt Status Register[5] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOFLW[31..0]	Read	0x0000	**Receive Overflow Interrupt[31..0]:** RXOFLW[0] corresponds to Receive Buffer 1 while RXOK[31] corresponds to Receive Buffer 32. Each bit indicates that the a valid message is present in the Receive Buffer has been overwritten by the controller. A Receive Buffer sees its contents as a valid message while the RXOK bit is '1'. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer has been overwritten. '0' = Indicates that the message present in the Receive Buffer has not been overwritten. Register Bits RXBISR[5][0..29] are invalid if "Number of Rx Buffers" is 2 Register Bits RXBISR[5][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBISR[5][0..23] are invalid if "Number of Rx Buffers" is 8 Register Bits RXBISR[5][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 178: **Receive Buffer Interrupt Status Register[6] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLW[63..32]	Read	0x0000	**Receive Overflow Interrupt[63..32]:** RXOFLW[32] corresponds to Receive Buffer 33 while RXOFLW[63] corresponds to Receive Buffer 64. Each bit indicates that the a valid message is present in the Receive Buffer has been overwritten by the controller. A Receive Buffer sees its contents as a valid message while the RXOK bit is '1'. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer has been overwritten. '0' = Indicates that the message present in the Receive Buffer has not been overwritten. This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 179: **Receive Buffer Interrupt Status Register[7] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOFLW[95..64]	Read	0x0000	**Receive Overflow Interrupt[95..64]:** RXOFLW[64] corresponds to Receive Buffer 65 while RXOFLW[95] corresponds to Receive Buffer 96. Each bit indicates that the a valid message is present in the Receive Buffer has been overwritten by the controller. A Receive Buffer sees its contents as a valid message while the RXOK bit is '1'. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer has been overwritten. '0' = Indicates that the message present in the Receive Buffer has not been overwritten. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 180: **Receive Buffer Interrupt Status Register[8] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLW[127..96]	Read	0x0000	**Receive Overflow Interrupt[127..96]:** RXOFLW[96] corresponds to Receive Buffer 97 while RXOFLW[127] corresponds to Receive Buffer 128. Each bit indicates that the a valid message is present in the Receive Buffer has been overwritten by the controller. There is a one to one correspondence between the bit position and the buffer number '1' = Indicates that a valid message is present in the Receive Buffer has been overwritten. '0' = Indicates that the message present in the Receive Buffer has not been overwritten. This register is valid only when "Number of Rx Buffers" is greater than 64.

Receive Buffer Interrupt Enable Register[1..8] (RXBIER[n])

The Receive Buffer Interrupt Enable Registers (`RXBIER[n]`) indicate the interrupt enable bits for the `RXOK` and `RXOFLW` bits in the `RXBISR[n]`. The number of Receive Buffer Interrupt Enable Registers and the number of bits in each of these registers depends on the parameter `Number of Rx Buffers`. Each `RXOK[x]` bit in the `RXBISR[n]` has a corresponding `RXOKE[x]` bit in the `RXBIER[n]`.

The address of Receive Buffer Interrupt Enable Registers can be computed as follows

Address of `RXBIER[n]` = `0x0000F60` + [n-1] × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Table 181 provides the RXBIER[1] description. Table 182 provides the RXBIER[2] description.

Table 183 provides the RXBIER[3] description. Table 184 provides the RXBIER[4] description.

Table 185 provides the RXBIER[5] description. Table 186 provides the RXBIER[6] description.

Table 187 provides the RXBIER[7] description. Table 188 provides the RXBIER[8] description.

Table 181: **Receive Buffer Interrupt Enable Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOKE[31..0]	Read/Write	0x0000	**Receive OK Interrupt Enable [31..0]**: RXOKE[0] corresponds to RXOK[0] bit in the RXBISR[1] while RXOKE[31] corresponds to RXOK[31] bit in the RXBISR[1]. '1' = When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is asserted when the RXOK[x] bit in the RXBISR[1] is '1.' '0' =When the RXOKE[x] bit is a '1',the Flex_Rxbirpt line is not asserted when the RXOK[x] bit in the RXBISR[1] is '1.' Register Bits RXBIER[1][0..29] are invalid if "Number of Rx Buffers" is 2. Register Bits RXBIER[1][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBIER[1][0..23] are invalid if "Number of Rx Buffers" is 8. Register Bits RXBIER[1][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 182: **Receive Buffer Interrupt Enable Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOKE[63..32]	Read/ Write	0x0000	**Receive OK Interrupt Enable [63..32]**: RXOKE[32] corresponds to RXOK[32] bit in the RXBISR[2] while RXOKE[63] corresponds to RXOK[63] bit in the RXBISR[2]. '1' = When the RXOKE[x] bit is a '1,' the Flex_Rxbirpt line is asserted when the RXOK[x] bit in the RXBISR[2] is '1.' '0' =When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is not asserted when the RXOK[x] bit in the RXBISR[2] is '1.' This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 183: **Receive Buffer Interrupt Enable Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOKE[95..64]	Read/ Write	0x0000	**Receive OK Interrupt Enable [95..64]**: RXOKE[64] corresponds to RXOK[64] bit in the RXBISR[3] while RXOKE[95] corresponds to RXOK[95] bit in the RXBISR[3]. '1' = When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is asserted when the RXOK[x] bit in the RXBISR[3] is '1.' '0' =When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is not asserted when the RXOK[x] bit in the RXBISR[3] is '1.' This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 184: **Receive Buffer Interrupt Enable Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOKE[127..96]	Read/Write	0x0000	**Receive OK Interrupt Enable [127..96]**: RXOKE[96] corresponds to RXOK[96] bit in the RXBISR[4] while RXOKE[127] corresponds to RXOK[127] bit in the RXBISR[4]. '1' = When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is asserted when the RXOK[x] bit in the RXBISR[4] is '1.' '0' =When the RXOKE[x] bit is a '1,'the Flex_Rxbirpt line is not asserted when the RXOK[x] bit in the RXBISR[4] is '1.' This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 185: **Receive Buffer Interrupt Enable Register[5] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOFLWE[31..0]	Read/ Write	0x0000	**Receive Overflow Interrupt Enable [31..0]:** RXOFLWE[0] corresponds to RXOFLW[0] bit in the RXBISR[5] while RXOFLWE[31] corresponds to RXOFLW[31] bit in the RXBISR[5]. '1' = When the RXOFLWE[x] bit is a '1,' the Flex_Rxbirpt line is asserted when the RXOFLW[x] bit in the RXBISR[5] is '1.' '0' =When the RXOFLWE[x] bit is a '1,' the Flex_Rxbirpt line is not asserted when the RXOFLW[x] bit in the RXBISR[5] is '1.' Register Bits RXBIER[5][0..29] are invalid if "Number of Rx Buffers" is 2. Register Bits RXBIER[5][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBIER[5][0..23] are invalid if "Number of Rx Buffers" is 8. Register Bits RXBIER[5][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 186: **Receive Buffer Interrupt Enable Register[6] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLWE[63..32]	Read/ Write	0x0000	**Receive Overflow Interrupt Enable [63..32]:** RXOFLWE[32] corresponds to RXOFLW[32] bit in the RXBISR[6] while RXOFLWE[63] corresponds to RXOFLW[63] bit in the RXBISR[6]. '1' = When the RXOFLWE[x] bit is a '1,'the Flex_Rxbirpt line is asserted when the RXOFLW[x] bit in the RXBISR[6] is '1.' '0' =When the RXOFLWE[x] bit is a '1,' the Flex_Rxbirpt line is not asserted when the RXOFLW[x] bit in the RXBISR[6] is '1.' This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 187: **Receive Buffer Interrupt Enable Register[7] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOFLWE[95..64]	Read/Write	0x0000	**Receive Overflow Interrupt Enable [95..64]**: RXOFLWE[64] corresponds to RXOFLW[64] bit in the RXBISR[7] while RXOFLWE[95] corresponds to RXOFLW[95] bit in the RXBISR[7]. '1' = When the RXOFLWE[x] bit is a '1','the Flex_Rxbirpt line is asserted when the RXOFLW[x] bit in the RXBISR[7] is '1.' '0' =When the RXOFLWE[x] bit is a '1,'the Flex_Rxbirpt line is not asserted when the RXOFLW[x] bit in the RXBISR[7] is '1.' This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 188: **Receive Buffer Interrupt Enable Register[8] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLWE[127..96]	Read/Write	0x0000	**Receive Overflow Interrupt Enable [127..96]**: RXOFLWE[96] corresponds to RXOFLW[96] bit in the RXBISR[8] while RXOFLWE[127] corresponds to RXOFLW[127] bit in the RXBISR[8]. '1' = When the RXOFLWE[x] bit is a '1',the Flex_Rxbirpt line is asserted when the RXOFLW[x] bit in the RXBISR[8] is '1'. '0' =When the RXOFLWE[x] bit is a '1',the Flex_Rxbirpt line is not asserted when the RXOFLW[x] bit in the RXBISR[8] is '1'. This register is valid only when "Number of Rx Buffers" is greater than 64.

Receive Buffer Interrupt Clear Register[1..8] (RXBICR[n])

The Receive Buffer Interrupt Clear Registers (`RXBICR[n]`) indicate the interrupt clear bits for the `RXOK` bits in the `RXBISR[n]`. The number of Receive Buffer Interrupt Clear Registers and the number of bits in each of these registers depends on the value set for the parameter `Number of Rx Buffers`. Each `RXOK[x]` bit in the `RXBISR[n]` has a corresponding `RXOKC[x]` bit in the `RXBICR[n]`.

The address of Receive Buffer Interrupt Clear Registers can be computed as follows

Address of `RXBICR[n]` = `0x0000F80` + [n-1] × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Reads are not allowed.

- Writes—Allowed in all operational states.

Table 189 provides the RXBICR[1] description. Table 190 provides the RXBICR[2] description.

Table 191 provides the RXBICR[3] description. Table 192 provides the RXBICR[4] description.

Table 193 provides the RXBICR[5] description. Table 194 provides the RXBICR[6] description.

Table 195 provides the RXBICR[7] description. Table 196 provides the RXBICR[8] description.

Table 189: **Receive Buffer Interrupt Clear Register[1] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOKC[31..0]	Write	0x0000	**Receive OK Interrupt Clear [31..0]**: RXOKC[0] corresponds to RXOK[0] bit in the RXBISR[1] while RXOKC[31] corresponds to RXOK[31] bit in the RXBISR[1]. '1' = Writing a '1' to the RXOKC[x] bit clears the RXOK[x] bit in the RXBISR[1]. Register Bits RXBICR[1][0..29] are invalid if "Number of Rx Buffers" is 2. Register Bits RXBICR[1][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBICR[1][0..23] are invalid if "Number of Rx Buffers" is 8. Register Bits TXBICR[1][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 190: **Receive Buffer Interrupt Clear Register[2] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOKC[63..32]	Write	0x0000	**Receive OK Interrupt Clear [63..32]:** RXOKC[32] corresponds to RXOK[32] bit in the RXBISR[2] while RXOKC[63] corresponds to RXOK[63] bit in the RXBISR[2]. '1' = Writing a '1' to the RXOKC[x] bit clears the RXOK[x] bit in the RXBISR[2]. This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 191: **Receive Buffer Interrupt Clear Register[3] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOKC[95..64]	Write	0x0000	**Receive OK Interrupt Clear [95..64]**: RXOKC[64] corresponds to RXOK[64] bit in the RXBISR[3] while RXOKC[95] corresponds to RXOK[95] bit in the RXBISR[3]. '1' = Writing a '1' to the RXOKC[x] bit clears the RXOK[x] bit in the RXBISR[3]. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 192: **Receive Buffer Interrupt Clear Register[4] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOKC[127..96]	Write	0x0000	**Receive OK Interrupt Clear [127..96]:** RXOKC[96] corresponds to RXOK[96] bit in the RXBISR[4] while RXOKC[127] corresponds to RXOK[127] bit in the RXBISR[4]. '1' = Writing a '1' to the RXOKC[x] bit clears the RXOK[x] bit in the RXBISR[4]. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 193: **Receive Buffer Interrupt Clear Register[5] Description**

Bits	Name	Access	Default Value	Field Description
0-31	RXOFLWC[31..0]	Write	0x0000	**Receive Overflow Interrupt Clear [31..0]:** RXOFLWC[0] corresponds to RXOFLW[0] bit in the RXBISR[5] while RXOFLWC[31] corresponds to RXOFLW[31] bit in the RXBISR[5]. '1' = Writing a '1' to the RXOFLWC[x] bit clears the RXOFLW[x] bit in the RXBISR[5]. Register Bits RXBICR[5][0..29] are invalid if "Number of Rx Buffers" is 2. Register Bits RXBICR[5][0..27] are invalid if "Number of Rx Buffers" is 4. Register Bits RXBICR[5][0..23] are invalid if "Number of Rx Buffers" is 8. Register Bits TXBICR[5][0..15] are invalid if "Number of Rx Buffers" is 16. All the register bits are valid if "Number of Rx Buffers" are greater than or equal to 32.

Table 194: **Receive Buffer Interrupt Clear Register[6] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLWC[63..32]	Write	0x0000	**Receive Overflow Interrupt Clear [63..32]**: RXOFLWC[32] corresponds to RXOFLW[32] bit in the RXBISR[6] while RXOFLWC[63] corresponds to RXOFLW[63] bit in the RXBISR[6]. '1' = Writing a '1' to the RXOFLWC[x] bit clears the RXOFLW[x] bit in the RXBISR[6]. This register is valid only when "Number of Rx Buffers" is greater than 32.

Table 195: **Receive Buffer Interrupt Clear Register[7] Description**

Bits	Name	Access	Default Value	Field Description
0- 31	RXOFLWC[95..64]	Write	0x0000	**Receive Overflow Interrupt Clear [95..64]**: RXOFLWC[64] corresponds to RXOFLW[64] bit in the RXBISR[7] while RXOFLWC[95] corresponds to RXOFLW[95] bit in the RXBISR[7]. '1' = Writing a '1' to the RXOFLWC[x] bit clears the RXOFLW[x] bit in the RXBISR[7]. This register is valid only when "Number of Rx Buffers" is greater than 64.

Table 196: **Receive Buffer Interrupt Clear Register[8] Description**

Bits	Name	Access	Default Value	Field Description
0 - 31	RXOFLWC[127..96]	Write	0x0000	**Receive Overflow Interrupt Clear [127..96]**: RXOFLWC[96] corresponds to RXOFLW[96] bit in the RXBISR[8] while RXOFLWC[127] corresponds to RXOFLW[127] bit in the RXBISR[8]. '1' = Writing a '1' to the RXOFLWC[x] bit clears the RXOFLW[x] bit in the RXBISR[4]. This register is valid only when "Number of Rx Buffers" is greater than 64.

Receive FIFO Interrupt Status Register (RXFISR)

The Receive FIFO Interrupt Status Register (RXFISR) indicates the Receive FIFO status. This includes the empty, full, and watermark (threshold) statuses of the FIFO.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not allowed.

Table 197 and shows the bit positions in the RXFISR and Table 198 provides the RXFISR description.

Table 197: **Receive FIFO Interrupt Status Register Bit Positions**

0-28	29	30	31
RESERVED	RXFFULL	RXFEMPTY	RXFTHLD

Table 198: **Receive FIFO Interrupt Status Register Description**

Bits	Name	Access	Default Value	Description
0-28	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
29	RXFFULL	Read	0	**Receive FIFO Full Interrupt**: This bit indicates the Receive FIFO Full status. '1' = Indicates that the Receive FIFO is full. '0' = Indicates that the Receive FIFO has not attained full status the last clear operation.
30	RXFEMPTY	Read	0	**Receive FIFO Empty Interrupt**: This bit indicates the Receive FIFO Empty status '1' = Indicates that the Receive FIFO is empty. '0' = Indicates that the Receive FIFO has not attained empty status since the last clear operation.
31	RXFTHLD	Read	0	**Receive FIFO Threshold Interrupt**: This bit indicates the Receive FIFO Threshold. '1' = Indicates that the Receive FIFO Threshold has been exceeded. '0' = Indicates that the Receive FIFO Threshold has not been exceeded since the last clear operation.

Receive FIFO Interrupt Enable Register (RXFIER)

The Receive FIFO Interrupt Enable Register (RXFIER) indicates the interrupt enable bits for the interrupt status bit in the RXFISR.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Table 199 and shows the bit positions in the RXFIER and Table 200 provides the RXFIER description.

Table 199: **Receive FIFO Interrupt Enable Register Bit Positions**

0-28	29	30	31
RESERVED	RXFFULLE	RXFEMPTYE	RXFTHLDE

Table 200: **Receive FIFO Interrupt Enable Register Description**

Bits	Name	Access	Default Value	Description
0-28	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion

Bits	Name	Access	Default Value	Description
29	RXFFULLE	Read	0	**Receive FIFO Full Interrupt Enable**: This bit is the enable bit for the RXFFULL bit in the RXFISR. '1' = When the RXFULLE bit is a '1,' the Flex_Rxfirpt line is asserted when the RXFULL bit in the RXFISR is '1.' '0' =When the RXFULLE bit is a '1, ' the Flex_RxFirpt line is not asserted when the RXFULL bit in the RXFISR is '1.'
30	RXFEMPTYE	Read	0	**Receive FIFO Empty Interrupt Enable**: This bit is the enable bit for the RXFEMPTY bit in the RXFISR. '1' = When the RXFEMPTYE bit is a '1,' the Flex_Rxfirpt line is asserted when the RXFEMPTY bit in the RXFISR is '1.' '0' =When the RXFEMPTYE bit is a '0,' the Flex_RxFirpt line is not asserted when the RXFEMPTY bit in the RXFISR is '1.'
31	RXFTHLDE	Read	0	**Receive FIFO Threshold Interrupt Enable**: This bit is the enable bit for the RXFTHDL bit in the RXFISR. '1' = When the RXFTHDLE bit is a '1,' the Flex_Rxfirpt line is asserted when the RXFTHDL bit in the RXFISR is '1.' '0' =When the RXFTHDLE bit is a '0,' the Flex_RxFirpt line is not asserted when the RXFTHDL bit in the RXFISR is '1.'

Receive FIFO Interrupt Clear Register (RXFICR)

The Receive FIFO Interrupt Clear Register (RXFICR) indicates the interrupt clear bits for the interrupt status bits in the RXFISR.

Access Criterion for Reads/Writes

- Reads—Not allowed.
- Writes—Allowed in all operational states.

Table 201 and shows the bit positions in the RXFICR and Table 202 provides the RXFICR description.

Table 201: **Receive FIFO Interrupt Clear Register Bit Positions**

0-28	29	30	31
RESERVED	RXFFULLC	RXFEMPTYC	RXFTHLDC

Table 202: **Receive FIFO Interrupt Clear Register Description**

Bits	Name	Access	Default Value	Description
0-28	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
29	RXFFULLC	Write	0	**Receive FIFO Full Interrupt Clear**: This bit is the clear bit for the RXFFULL bit in the RXFISR. '1' = Writing a '1' to the RXFFULLC bit clears the RXFFULL bit in the RXFISR.
30	RXFEMPTYC	Write	0	**Receive FIFO Empty Interrupt Clear**: This bit is the clear bit for the RXFEMPTY bit in the RXFISR. '1' = Writing a '1' to the RXFEMP-TYC bit clears the RXFEMPTY bit in the RXFISR.
31	RXFTHLDC	Write	0	**Receive FIFO Threshold Interrupt Clear:** This bit is the clear bit for the RXFTHLD bit in the RXFISR. '1' = Writing a '1' to the RXFTHLDC bit clears the RXFTHLD bit in the RXFISR.

Transmit Buffers

The number of Transmit buffers in the FlexRay controller can be configured based on the parameter `Number of Tx Buffers`. Each Transmit buffer consists of three dedicated header registers. The header registers contain control information (such as Slot ID, Buffer Validity bit, Pointer to Payload Storage). The storage of data associated with a Transmit buffer is done in the Payload buffer.

Payload Buffer

The storage allocated for the payload portion of the Transmit buffer can vary from 2 KBytes to 8 KBytes in increments of 2 KBytes. You must determine the amount of storage allocated to each Transmit buffer. It is recommended that the size of memory in the Payload Buffer allocated to each Transmit Buffer be `Max Payload Size`.

During frame transmission, the `PE` reads data from the Payload buffer at the starting address indicated by the `TPADDR` field in `TXBH3R[n]`. The last address at which the PE reads the data is indicated by `TPADDR` field in the `TXBH3R[n]` + `SSPLR` (`gPayloadLengthStatic`) for Static Frames and `TPADDR` field in `TXBH3R[n]`+`TPLEN` field in `TXBH2R[n]` for Dynamic Frames.

Read/Write burst access to Payload buffer is supported.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—For a section of Payload buffer allocated to a `Transmit Buffer[n]`, the locations in the Payload buffer section can be updated only when the `TBNA` bit of the `Transmit buffer[n]` is a '0.'

Table 203 shows the bit positions for a single memory location in the Payload buffer and Table 204 provides the Payload buffer description.

Table 203: **Payload Buffer Bit Positions**

0-7	8-15	16-23	24-31
TDB0[7..0]	TDB1[7..0]	TDB2[7..0]	TDB3[7..0]

Table 204: **Payload Buffer Description**

Bits	Name	Access	Default Value	Description
0-7	TDB0[7..0]	Read/Write	0	Transmission Data Byte 0. 1st byte of payload data.
8-15	TDB1[7..0]	Read/Write	0	Transmission Data Byte 1. 2nd byte of payload data.
16-23	TDB2[7..0]	Read/Write	0	Transmission Data Byte 2. 3rd byte of payload data.
24-31	TDB3[7..0]	Read/Write	0	Transmission Data Byte 3. 4th byte of payload data.

DB0 is the first byte to be transmitted, and DB3 is the last byte to be transmitted from a Payload buffer memory location during frame transmission.

Transmit Buffer Header 1 Registers[1..128] (TXBH1R[n])

The number of Transmit Buffer Header 1 (`TXBH1R[n]`) registers are equal to the parameter `Number of Tx Buffers`.

Transmit Buffer Header 1 register specifies the `Buffer Not Available` bit.

The address of Transmit Buffer Header 1 Register of a particular Transmit buffer can be computed as:

Address of TXBH1R[n] = 0x00003000 + (n-1) × 0x00000010

For example, the address of TXBH1R[3] can be computed as:

Address of TXBH1R[3] = 0x00003000 + 2 × 0x00000010 = 0x00003030

Read/Write burst access to this register is not supported.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Write to clear.

Writes: Write to clear.

Table 205 shows the bit positions in the TXBH1R[n] and Table 206 provides the TXBH1R[n] description.

Table 205: **Transmit Buffer Header 1 Register[n] Bit Position**

0-30	31
RESERVED	TBNA

Table 206: **Transmit Buffer Header 1 Register[n] Bit Field Description**

Bits	Name	Access	Default Value	Description
0-30	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
31	TBNA	Read/Write	0	**Transmission Buffer Not Available:** '1' = Indicates that frame present in the Transmit buffer is to be transmitted by the node. Also, indicates that the Transmit buffer is locked for access by the PE. '0'= Indicates that the frame present in the Transmit buffer and the payload data associated with the Transmit buffer is not be transmitted by the node. In Static segment, a Null Frame is sent during the slot indicated by the Transmit buffer and in Dynamic segment, no frame transmission takes place. Also, indicates that the Transmit buffer is locked for access by the host. This bit is set to '0' provided the TSS bit in TXBH2R[n] is set to '0'. In Startup State, the TBNA bits are not cleared.' Writing a '0' to this bit clears this bit, provided the message in the buffer is not currently being transmitted. In such a case, this bit continues to remain '1.'

Transmit Buffer Header 2 Registers [1..128] (TXBH2R[n])

The number of Transmit Buffer Header 2 (TXBH2R[n]) registers is equal to the parameter Number of Tx Buffers. Transmit Buffer Header 2 Register specifies the Network Management Vector Indicator bit, Single Shot bit, the Frame ID, Payload length and Header CRC fields.

The address of Transmit Buffer Header 2 Register of a particular Transmit buffer can be computed as:

Address of TXBH2R[n] = 0x00003004 + (n-1) × 0x00000010

For example, the address of TXBH2R[3] can be computed as:

Address of TXBH2R[3] = 0x00003004 + 2 × 0x00000010 = 0x00003034

Read/Write burst access to this register is not supported.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—TSS, TFID can be written to only in Config state. All other fields can be written to only when TBNA bit in the TXBH1R[n] is a '0.'

Table 207 shows the bit positions in the TXBH2R[n] and Table 208 provides the TXBH2R[n] description.

Table 207: **Transmit Buffer Header 2 Register[n] Bit Positions**

0	1	2	3-13	14-20	21-31
Reserved	TSS	TNMI	TFID(10:0)	TPLEN(6:0)	THCRC(10:0)

Table 208: **Transmit Buffer Header 2 Register[n] Description**

Bits	Name	Access	Default Value	Description
0	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
1	TSS	Read/Write	0	**Transmission Single Shot**: This bit indicates whether the message in the Transmit buffer is sent once (single shot) or on a continuous basis '1' = The message in the Transmit buffer is transmitted only once. '0' = The message in the Transmit buffer is transmitted every cycle. This field can be written to only in Config state. The header registers and payload memory in the pay-load buffer cannot be updated by the host when the TSS bit is set to '0.'
2	TNMI	Read/Write	0	**Transmission Network Management Vector Indicator**: This bit indicates the presence of a Network Management Vector / Message ID in the Transmit buffer. '1' = The payload in the Transmit buffer contains a Network Management Vector for a Static Frame or a Message ID for a Dynamic Frame. '0' = The payload in the Transmit buffer does not contain a Network Management Vector or a Message ID. This bit can be written to only when the TBNA bit in TBH1R[n is '0.'

Bits	Name	Access	Default Value	Description
3-13	TFID(10:0)	Read/Write	0	**Transmission Frame ID**: This field indicates the slot during which the message in the Transmit buffer should be transmitted. This field can be written to only in Config state of operation.
14-20	TPLEN(6:0)	Read/Write	16	**Transmission Payload Length**: Indicates the length of the payload for the message in the Transmit buffer. This field can be written to only when the TBNA bit in TBH1R[n is '0.'
21-31	THCRC(10:0)	Read/Write	0	**Transmission Header CRC:** Indicates the Header CRC for the message in the Transmit buffer This field can be written to only when the TBNA bit in TBH1R[n is '0.'

Transmit Buffer Header 3 Registers [1..128] (TXBH3R[n])

The number of Transmit Buffer Header 3 (`TXBH3R[n]`) registers are equal to the Number of Tx Buffers parameter. Transmit Buffer Header 3 register specifies the payload start address pointer. It indicates start address of the payload for the corresponding Transmit buffer.The value to be written into `TPADDR` is should be 0x"00001000" + Payload address offset.

The address of Transmit buffer Header 3 Register of a particular Transmit buffer can be computed as follows:

Address of `TXBH3R[n]` = `0x00003008` + (n-1) × `0x00000010`

For Example, the address of `TXBH3R[3]` can be computed as:

Address of `TXBH3R[3]` = `0x00003008` + 2 × `0x00000010` = `0x00003038`

Read/Write burst access to this register is not supported.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Can be written to only when `TBNA` bit in the `TXBH1R[n]` is a '0'.

Table 209 shows the bit positions in the TXBH3R[n] and Table 210 provides the TXBH3R[n] description.

Table 209: **Transmit Buffer Header 3 Register[n] Bit Positions**

0-19	18-31
RESERVED	TPADDR(13:0)

Table 210: **Transmit Buffer Header 3 Register[n] Description**

Bits	Name	Access	Default Value	Description
0-19	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion
18-31	TPADDR(13:0)	Read/Write	0	**Transmission Payload Address**: Indicates the pointer for the payload data stored in the payload buffer in FlexRay controller. This field can be written to only when the TBNA bit in TBH1R[n is '0.' The Transmission Payload Address indicates the byte address. TPADDR(1:0) should always be '0s' to align the address to a 32 bit boundary.

Receive Buffers

The number of Receive buffers in the FlexRay controller is configured with the `Number of Rx Buffers` parameter. Each Receive buffer consists of up to 66 dedicated memory locations.

Receive Buffer [1..66] Register [1..128] (RXB[y]R[n])

Receive Buffer [1..66] Register [1..128] (`RXB[y]R[n]`) indicates y^{th} register in the n^{th} Receive Buffer. The number of Receive Buffer Registers depend on the `Number of Rx Buffers` and `Maximum Size of Payload` parameters.

- Y can take values from 1 to 66. y can be computed as (2 + `Max Size of Payload`/4).
- N can take values from 1 to `Number of Rx Buffers`.

`RXB1R[n]` and `RXB2R[n]` provide the header information for the received message stored in `Rx Buffer[n]`. The registers `RXB3R[n]` to `RXB[y]R[n]` provide the payload information for the received message. The address mapping for the `RXB[y]R[n]` registers varies based on the `Maximum Size of Payload` parameter.

The starting address of the Register set for a particular `Receive Buffer[n]` can be computed as follows (*n* indicates the Receive Buffer number):

Address of `RXB1R[n]` = `0x00004000` + (n-1) × (2 + `Max Payload Size`/4) + (y-1) × `0x00000004`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Not Allowed.

Table 211 shows the bit positions in the RXB1R[n] and Table 212 provides the RXB1R[n] description.

Table 211: **Receive Buffer 1 Register [n] Bit Positions**

0	1	2	3	4	5-15	16-22	23-31
RXRSVD	RXPPI	RXNFI	RXSYFI	RXSUFI	RXFID(10:0)	RXPLEN(6:0)	RXHCRC(8:0)

Table 212: **Receive Buffer 1 Register [n] Bit Description**

Bits	Name	Access	Default Value	Description
0	RXRSVD	Read/Write	0	**Received Reserved Bit**: Indicates the Reserved bit.
1	RXPPI	Read	0	**Received Payload Preamble Bit:** Indicates the Payload preamble. '1'= If the frame is received in Static segment, then payload consists of Network Management Vector. '1'= If the frame is received in the dynamic segment then message ID is available in the beginning of the payload. '0' = If frame is received in Static segment, then Network Management Vector is not present. I'0' = If frame is received in the dynamic segment then message ID is not present.
2	RXNFI	Read	1	**Received Null Frame Indicator:** If set to '1,' payload consists valid data. If set to '0,' payload does not consist any valid data. All bytes are zero.
3	RXSYFI	Read	0	**Received Sync Frame Indicator:** If set to '1,' the frame is a Sync Frame. If set to '0,' the frame is not a Sync Frame.
4	RXSUFI	Read	0	**Received Startup Frame Indicator:** If set to '1,' the frame is a Startup Frame. If set to '0,' the frame is not a Startup Frame.

Bits	Name	Access	Default Value	Description
5-15	RXFID(10:0)	Read	0x000	**Received Frame ID:** Indicates the slot number where the frame is received.
16-22	RXPLEN(6:0)	Read	0x00	**Received Payload Length:** Payload Length of frame.
23-31	RXHCRC(8:0)	Read	0x000	**Received Header CRC:** Header CRC of a Frame.

Table 213 shows the bit positions in the RXB2R[n] and Table 214 provides the RXB2R[n] description.

Table 213: **Receive Buffer 2 Register [n] Bit Positions**

0-1	2-7	8-31
RXHCRC(10:9)	RXCC(5:0)	Reserved

Table 214: **Receive Buffer 2 Register [n] Bit Description**

Bits	Name	Access	Default Value	Description
0-1	RXHCRC(10:9)	Read	0x0	**Received Header CRC**: Header CRC of a Frame.
2-7	RXCC(5:0)	Read	0x00	**Received Cycle Count:** Cycle Count field of Frame.
8-31	Reserved	Read	0x00	**Reserved**: These bit positions are reserved for future use.

Table 215 shows the bit positions in the RXB[y]R[n] and Table 216 provides the RXB[y]R[n] description.

Table 215: **Receive Buffer [y] Register[n] Bit Positions**

0-15	16-31
PAYLOADWORD1(15:0)	PAYLOADWORD2(15:0)

Table 216: **Receive Buffer [y] Register[n] Bit Description**

Bits	Name	Access	Default Value	Description
0-15	PAYLOADWORD1(15:0)	Read	0x00	**Received Data Payload Word 1**: Payload Data word1
16-31	PAYLOADWORD2(15:0)	Read	0x00	**Received Data Payload Word 2**: Payload Data word2

Receive FIFO

The depth of Receive FIFO in the FlexRay controller can be configured with the Depth of Rx FIFO parameter. Each message in the Receive FIFO can accept up to 66 memory locations.

Receive FIFO[1..66] Registers (RXF[y]R)

Receive FIFO[1..66] Registers (`RXF[y]R`) indicates the y location of a message in the Receive FIFO. The number of locations in a received message in the Receive FIFO depends on the Number of Rx Buffers and Max Size of Payload parameters. Y can accept values from 1 to 66 and can be computed as:

(2 + `Max Payload Size`/4)

The structure of a message in the Receive FIFO is the same as that of a message stored in the Receive buffers. The Receive FIFO is assigned an address space of 256 bytes (*y* indicates the register number):

Address of `RXF[y]R` = `0x00006000` + (y-1) × `0x00000004`

Each message in the Receive FIFO is allocated a storage space of (8+ `Maximum Size of Payload`). To read a complete message, the number of reads performed on the FIFO should be equal to (2 + (`Maximum Size of Payload`/4)), irrespective of the payload size of the message stored in the Receive FIFO.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states. Reads are not allowed when the FIFO is empty.
- Writes—Not allowed.

The bit positions for messages in the Receive FIFO and the Receive Buffers are the same.

Hence Table 211 shows the bit positions in the RXF1R and Table 212 provides the RXF1R description. Table 213 shows the bit positions in the RXF2R and Table 214 provides the RXF2R description. Table 215 shows the bit positions in the RXF[y]R and Table 216 provides the RXF[y]R description.

Receive FIFO Threshold Register (RFTR)

The Receive FIFO Threshold Register (`RFTR`) indicates the FIFO threshold reference level. When the number of messages stored in Receive FIFO exceeds the threshold value specified in the `RFTR`, the `RXFTHLD` bit in `RXFISR` is set.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed in all operational states.

Table 217 shows the bit positions in the RFTR and Table 218 provides the RFTR description.

Table 217: **Receive FIFO Threshold Register Bit Positions**

0-24	25-31
RESERVED	FIFOTHLDSET(6:0)

Table 218: **Receive FIFO Threshold Register Description**

Bits	Name	Access	Default Value	Description
0-24	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
25-31	FIFOTHLDSET(6:0)	Read/Write	0x0	**Receive FIFO Threshold**: This register acts as a reference against which the number of messages in the Receive FIFO is compared. The value in this register should always be less than the parameter "Rx FIFO Depth."

Acceptance Filters

Each Receive Buffer is associated with a Receive Buffer Acceptance Filter Register pair (`RBAF[n]`). A Receive Buffer Acceptance Filter Register pair is made up of Receive Buffer Acceptance Filter 1 Register and Receive Buffer Acceptance Filter 2 Register.

The Receive FIFO is associated with four pairs of Receive FIFO Acceptance Filters (`RFAF[n]`). Each Maskable Acceptance Filter consists of an ID Register and a Mask Register.

The following algorithm is used in computing filter pass/fail for storage into the Rx FIFO.

((Rx Field) & (Field Mask)) == ((Field data) & (Field Mask))

Where '&' is a bitwise logical 'AND' operation.

Receive Buffer Acceptance Filter 1 Register[n] (RBAF1R[n])

The RBAF1R[n] stores the Cycle Count, Cycle Mask and Frame IDs along with the respective filter field enable bits for Receive Buffer[n].

The address of Receive Buffer Acceptance Filter 1 Register that is associated with a particular Receive Buffer can be computed as:

$$\text{Address of } \texttt{RBAF1R[n]} = \texttt{0x00006300} + (n-1) \times \texttt{0x00000008}$$

where n indicates the Receive Buffer number.

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 219 shows the bit positions in the RBAF1R[n] and Table 220 provides the RBAF1R[n] description.

Table 219: **Receive Buffer Acceptance Filter 1 Register[n] Bit Position**

0-5	6	7	8-13	14-19	20	21-31
RESERVED	FEN	CCE	CCM(5:0)	CC(5:0)	FIDE	FID(10:0)

Table 220: **Receive Buffer Acceptance Filter 1 Register[n] Bit Field Description**

Bits	Field ID	Access	Default Value	Description
0-5	RESERVED	Read/ Write	0	**Reserved**: These bit positions are reserved for future expansion.
6	FEN	Read/ Write	0	**Filter Enable**: This bit enables the usage of RBAF1R[n] and RBAF2R[n] registers for the receive filtering. '1' = Receive Filtering is enabled for Receive Buffer. '0' = Receive Buffer is disabled for Receive Buffer.
7	CCE	Read/ Write	0	**Cycle Count Enable**: This bit enables the usage of the Cycle Count field in the RBAF1R[n] for filtering. '1' = Enables the usage of Cycle Count for frame filtering. '0' = Disables the usage of Cycle Count for frame filtering.
8-13	CCM(5:0)	Read/ Write	0x01	**Cycle Count Mask**: Indicates the Cycle Count mask value that is used for frame filtering.
14-19	CC(5:0)	Read/ Write	0x01	**Cycle Count Field:** Indicates the Cycle Count value that is used for frame filtering.
20	FIDE	Read/ Write	0	**Frame ID Enable**: This bit enables the usage of the Frame ID field in the RBAF1R[n] for filtering. '1' = Enables the usage of Frame ID for frame filtering. '0' = Disables the usage of Frame ID for frame filtering.
21-31	FID (10:0)	Read/Write	0	**Frame ID Field**: This field indicates the Frame ID value that is used for frame filtering.

Receive Buffer Acceptance Filter 2 Register[n] (RBAF2R[n])

The VReceive Buffer Acceptance Filter 2 Register[n] (RBAF2R[n]) stores the Message ID along with the filter enable bit for the Receive Buffer[n].

The address of RBAF2R[n] that is associated with a particular Receive Buffer can be computed as follows (*n* indicates the Receive Buffer number):

Address of RBAF2R[n] = 0x00006304 + (n-1) × 0x00000008

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 221 shows the bit position in the RBAF2R[n] and Table 222 provides the RBAF2R[n] description.

Table 221: **Receive Buffer Acceptance Filter 2 Register[n] Bit Positions**

0-14	15	16-31
RESERVED	MIDE	MID(15:0)

Table 222: **Receive Buffer Acceptance Filter 2 Register[n] Bit Description**

Bits	Field ID	Access	Default Value	Description
0-14	RESERVED	Read/ Write	0	**Reserved**: These bit positions are reserved for future expansion..
15	MIDE	Read/ Write	0	**Message ID Enable**: This bit enables the usage of the Message ID field in the RBAF2R[n] for filtering. '1' = Enables the usage of Message ID for frame filtering. '0' = Disables the usage of Message ID for frame filtering.
16-31	MID(15:0)	Read/ Write	0x01	**Message ID Field**: This field indicates the Message ID value that is used for frame filtering.

Receive FIFO Acceptance Filter ID 1 Register[1..4] (RFAFI1R[n])

The Receive FIFO Acceptance Filter ID 1 Register[1..4] (RFAFI1R[n]) contains the Cycle count and Frame ID fields. Four such ID registers exist for the FlexRay controller.

The address of RFAFI1R[n] can be computed as follows:

Address of RFAFI1R[n] = 0x00006700 + (n-1) × 0x00000010

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 223 shows the bit positions in the RFAFI1R[n] and Table 224 provides the RFAFI2R[n] description.

Table 223: **Receive FIFO Acceptance Filter ID 1 Register [n] Bit Positions]**

0-14	15-20	21-31
RESERVED	CC(5:0)	FID(10:0)

Table 224: **Receive FIFO Acceptance Filter ID 1 Register [n] Bit Description**

Bits	Name	Access	Default Value	Description
0-14	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
15-20	CC(5:0)	Read/Write	0	**Cycle Count Field**: This field indicates the Cycle count value that is used for frame filtering.
21-31	FID(10:0)	Read/Write	0	**Frame ID Field**: This field indicates the Frame ID value that is used for frame filtering.

Receive FIFO Acceptance Filter Mask 1 Register [1..4] (RFAFM1R[n])

The Receive FIFO Acceptance Filter Mask 1 Register [1..4] (`RFAFM1R[n]`) registers contain the Cycle count mask and Frame ID mask fields. Four such ID registers exist for the FlexRay controller.

The address of `RFAFM1R[n]` can be computed as:

Address of `RFAFM1R[n]` = `0x00006704` + (n-1) × `0x00000010`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 225 shows the bit positions in the RFAFM1R[n] and Table 226 provides the RFAFM1R[n] description.

Table 225: **Receive FIFO Acceptance Filter Mask 1 Register [n] Bit Position]**

0-14	15-20	21-31
RESERVED	CCM(5:0)	FIDM(10:0)

Table 226: **Receive FIFO Acceptance Filter Mask 1 Register [n] Bit Description**

Bits	Name	Access	Default Value	Description
0-14	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
15-20	CCM(5:0)	Read/Write	0	**Cycle Count Mask**: These bits are used to mask the Cycle count value in RFAFI1R[n] registers. '1' = Indicates that the corresponding Cycle count bit in the RFAFI1R[n] is used to for acceptance filtering. '0' = Indicates that the corresponding Cycle count bit in the RFAFI1R[n] is not used for acceptance filtering.
21-31	FIDM(10:0)	Read/Write	0	**Frame ID Mask**: These bits are used to mask the Frame ID value in RFAFI1R[n] registers '1' = Indicates that the corresponding Frame ID bit in the RFAFI1R[n] register is used to for acceptance filtering. '0' = Indicates that the corresponding Frame ID bit in the RFAFI1R[n] register is not used for acceptance filtering.

Receive FIFO Acceptance Filter ID 2 Register [1..4] (RFAFI2R[n])

The Receive FIFO Acceptance Filter ID 2 Register [1..4] (`RFAFI2R[n]`) contain the Message ID field. Four such ID registers exist for the FlexRay controller.

The address of `RFAFI2R[n]` can be computed as:

Address of `RFAFI2R[n]` = `0x00006708` + (n-1) × `0x00000010`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 227 shows the bit positions in the RFAFI2R[n] and Table 228 provides the RFAFI2R[n] description.

Table 227: **Receive FIFO Acceptance Filter ID 2 Register [n] Bit Positions]**

0-15	16-31
RESERVED	MID(15:0)

Table 228: **Receive FIFO Acceptance Filter ID 2 Register [n] Description**

Bits	Name	Access	Default Value	Description
0-15	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
16-31	MID(15:0)	Read/Write	0	**Message ID Field**: This field indicates the Message ID value that is used for frame filtering.

Receive FIFO Acceptance Filter Mask 2 Register [1..4] (FAFM2R[n])

The Receive FIFO Acceptance Filter Mask 2 Register [1..4] (`RFAFM2R[n]`) registers contain the Message ID mask field. Four such ID registers exist for the FlexRay controller.

The address of `RFAFM2R[n]` can be computed as:

Address of `RFAFM2R[n]` = `0x0000670C` + (n-1) × `0x00000010`

Access Criterion for Reads/Writes

- Reads—Allowed in all operational states.
- Writes—Allowed only in Config state.

Table 229 shows the bit positions in the RFAFM2R[n] and Table 230 provides the bit field description for RFAFM2R[n] description.

Table 229: **Receive FIFO Acceptance Filter Mask 2 Register [n] Bit Positions]**

0-15	16-31
RESERVED	MIDM(15:0)

Table 230: **Receive FIFO Acceptance Filter Mask 2 Register [n] Bit Description**

Bits	Name	Access	Default Value	Description
0-15	RESERVED	Read/Write	0	**Reserved**: These bit positions are reserved for future expansion.
16-31	MIDM(15:0)	Read/Write	0	**Message ID Mask**: These bits are used to mask the Message ID value in RFAFI2R[n] registers. '1' = Indicates that the corresponding MessageID bit in the RFAFI2R[n] register is used to for acceptance filtering. '0' = Indicates that the corresponding Message ID bit in the RFAFI2R[n] register is not used for acceptance filtering.

PLB Transactions

The external host interface of the FlexRay node is a 32-bit PLB bus. The FlexRay controller supports the following modes of data transfer:

- Single Read/Write
- Burst Read/Write

Single Read/Write

Single reads and single writes are supported over the entire address map.

PLB Single Read

Figure 3 shows the timing interface for single read transfer. Sl_addrAck, Sl_rdDAck and Sl_rdComp must be asserted within 16 PLB clock cycles after assertion of PLB_PAValid (including the clock cycle where the PLB_PAValid is asserted).

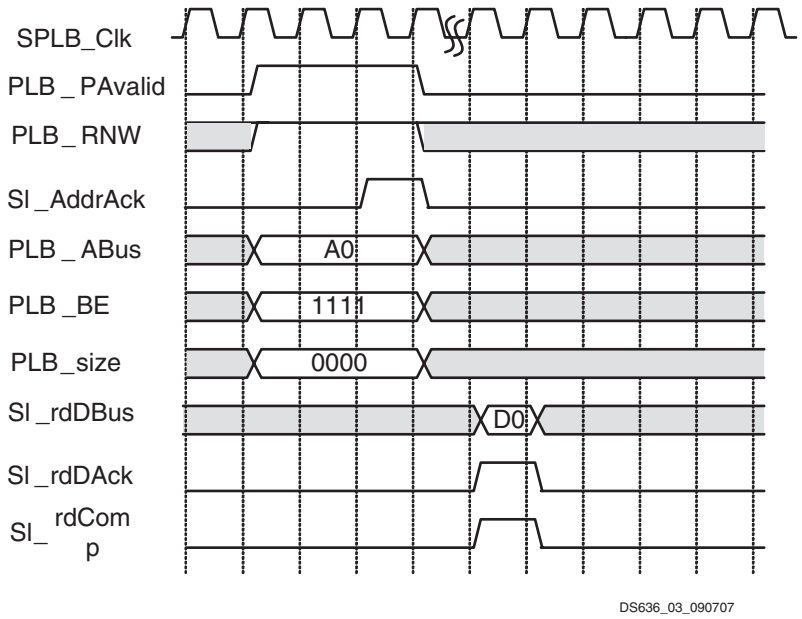

Figure 3: **PLB Single Read Transaction**

PLB Single Write

Figure 4 shows the timing interface for single write transfer. Sl_addrAck, Sl_wrDAck and Sl_wrComp must be asserted within 16 PLB clock cycles after assertion of PLB_PAValid (including the clock cycle where the PLB_PAValid is asserted).

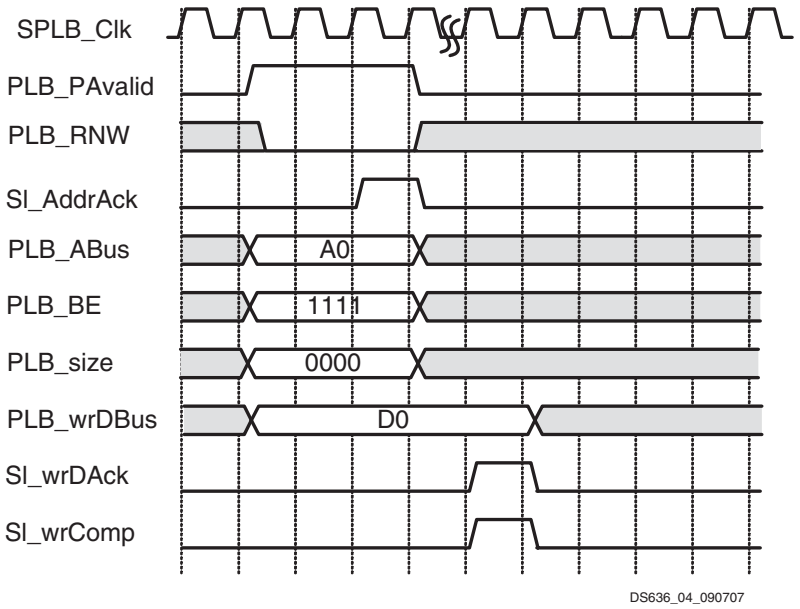

Figure 4: **PLB Single Write Transaction**

Burst Read/Write

Burst reads are supported over the address range `0x00001000-0x00002FFC` and `0x00004000-0x00005FFC`. Burst Writes are supported over the address range `0x00001000-0x00002FFC`. XPS FlexRay supports burst transactions of length 2, 4, 8, 12, 13, and 16. Burst operations starting in non-burstable address region and ending in burstable address region are not allowed. Burst operations starting in burstable address region and ending in non-burstable address region are not allowed.

PLB Burst Read

Figure 5 shows the timing interface for burst read transfer. Sl_addrAck must be asserted within 16 PLB clock cycles after the assertion of PLB_PAValid (including the clock cycle where the PLB_PAValid is asserted).

This timing diagram shows throttled burst read transaction. However, throttled burst is allowed where the data need not be provided in each clock. In this case, DUT may take several clock cycles between two data.

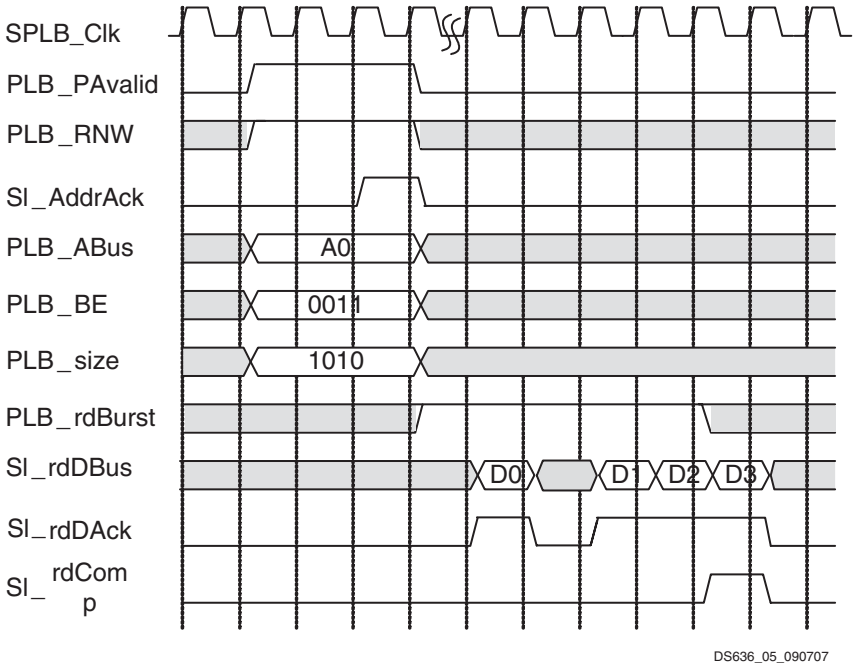

Figure 5: **PLB Burst Read Transaction**

PLB Burst Write

Figure 6 shows the timing interface for burst write transfer. Sl_addrAck must be asserted within 16 PLB clock cycles after the assertion of PLB_PAValid, including the clock cycle where the PLB_PAValid is asserted.

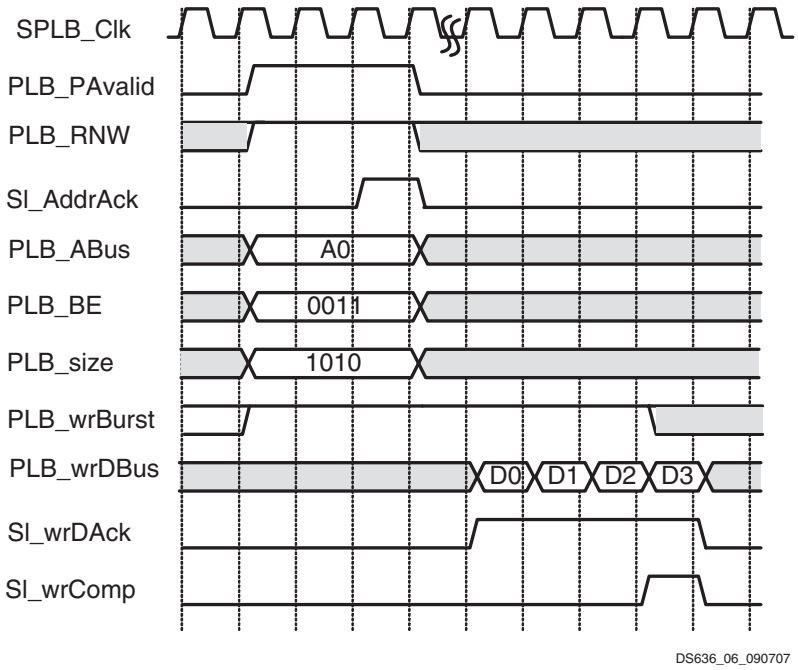

Figure 6: **PLB Write Burst Transaction**

Programming the FlexRay Core

This chapter provides information about the various configuration steps to program the FlexRay core. The following key configuration steps are detailed in this chapter:

- Halting the Controller
- Protocol Configuration Registers
- Configuring Transmit Buffers
- Configuring Receive Buffers/ Receive FIFO
- Absolute and Relative Timers

Halting the Controller

The controller ceases normal communication on the FlexRay bus when any of the following commands are asserted.

- **Software Reset**. Current frame reception or transmission is abruptly stopped and the controller enters the Reset state.
- **Hardware Reset**. Current frame reception or transmission is abruptly stopped and the controller enters the Reset state.

- **CHI_FREEZE command**. Current frame reception or transmission is abruptly stopped and the controller enters the Halt state.
- **CHI_HALT command**. The controller waits until the end of the current communication cycle and enters the Halt state.

Protocol Configuration Registers

The *FlexRay Protocol Specification Rev 2.1 A* defines the parameters required to ensure proper operation of a node within a cluster. These parameters are grouped into global-cluster and node-specific parameters.

- Global-cluster parameters must be programmed to the same value for all the nodes in the cluster.
- Node-specific parameters need not be programmed to the same value for all the nodes in the cluster.

The FlexRay controller defines a number of programmable registers that are based on the protocol specification. For better usability, the FlexRay controller registers that are based on the protocol-specified parameters have been grouped according to the internal controller sub-modules that use these parameters.

Protocol-Parameter Registers Sub-Categories

Registers that are protocol-parameter specific are grouped into the following sub-categories:

- Protocol Engine
- POC Module
- CODEC Module
- MAC Module
- Clock Sync Module
- Wakeup Module

Cluster-Level Parameters

To bring up a node in the FlexRay cluster, the following key parameters have to be configured at the cluster level:

- Duration of the cycle (indicated by the `NMICR` and `NMACR`)
- Duration of the static segment (indicated by the `SSDR` and `NSSR`)
- Duration of the dynamic segment (indicated by the `MSDR`, `NMSR`)
- Duration of the symbol window (indicated by the `SWDR`)
- Number of slots in the static segment (indicated by the `NSSR`)
- Number of slots in the dynamic segment (indicated by the `NMSR`)
- Action point offsets (indicated by the `APOR`, `MAPR`)
- Symbol related parameters (indicted by the `WRIR, WRLR, WRWR, WTIR, WTLR, CRMIR, CRMXR`)
- Maximum length of the payload (`SSPLR`)

Configuration of these parameters depends upon the number of nodes in the cluster, functionality, and bandwidth requirements for each node. Formulas for computing these parameters can be found in "Appendix B" of the *FlexRay Protocol Specification Rev 2.1 A*.

Node-Specific Parameters

After the cluster-level parameters are configured, then you configure the node-specific parameters. These parameters include:

- Static slots assigned (indicated by the `TXBH2R[n]`)
- Mini slots assigned (indicated by the `TXBH2R[n]`)
- Error handling characteristics (`AHCR, MCCFR, MCCPR, PAPAR`)
- Startup capability (indicated by the `STFCR, SYFCR, SYSR`)
- Wakeup-related parameters (indicated by the `LTOR, LNR, NCAR, WPSR`)
- Clock-related parameters, such as rate and offset corrections (indicated by the `EOCR`, `ERCR, OCLR, RCLR, MDR, CDDR, OCSR, MAOR, MIOR`).

Formulas for computing these parameters are found in "Appendix B" of the *FlexRay Protocol Specification Rev 2.1 A*.

Configuring Transmit Buffers

The FlexRay controller provides up to 128 transmit buffers. Transmit storage is organized into two separate sections: a transmit header storage area, and a payload storage area. The transmit header storage is divided into three-word (32 bytes) portions. Each three-word portion is associated with a specific transmit buffer. The header storage area contains the header and other control information for a transmitted frame.

The payload storage is a contiguous area of memory that stores the payload associated with each transmit buffer. The user must determine the size of the payload storage allocated to each `Transmit Buffer[n]`. The address offset of the start of payload data for associated with a particular transmit buffer is written to the TPADDR field of the Tx buffer header registers (TXBH3R).

Slot Assignment

The slot assignments for each of the nodes in the cluster must be determined during cluster design and configuration. The following guidelines must be adhered to ensure proper operation of the controller.

- Assign each FlexRay controller transmit buffer to a single slot. Multiple transmit buffers cannot be assigned to the same slot.
- The slot number (Frame ID) assigned to `Transmit Buffer [n]` should be written to the `TFID` field in the `TXBH2R[n]`. The TFID field of the `TXBH2R` can be written to only when the controller is in the Config state of operation.
- Slot numbers (Frame IDs) should be assigned to transmit buffers in the order of transmission. The first transmit buffer should always be assigned the lowest Frame ID that is transmitted by the controller, and the last transmit buffer that is configured should be assigned to the highest Frame ID. The intermediate transmit buffers should be assigned Frame IDs in the ascending order of Frame IDs. For example, if the host wants to transmit in slots 34, 145, 266, the slot numbers should be assigned to ascending transmit buffers in that same order.
- The configured transmit buffers must always be contiguous. Unconfigured transmit buffers between two configured transmit buffers is not allowed. This implies that the first transmit buffer configured must always be Transmit Buffer 1 and the last transmit buffer configured must be `Transmit Buffer [n]`, where *n* indicates the number of slots assigned to the controller.

  For example, If the host wants to transmit in slots 34, 145, 266, the slot assignments would be:

  slot number 34 is assigned to `Transmit Buffer [1]`

slot number 145 is assigned to `Transmit Buffer [2]`

slot number 266 is assigned to `Transmit Buffer [3]`

Transmit Buffer Selection

At the beginning of every communication cycle, the buffer selection pointer is initialized to point to `Transmit Buffer [1]`. In each slot, the slot number indicated by the `TFID` field in the `TXBH2R` of the transmit buffer is compared with the current slot number. When they match the buffer is evaluated for transmission, and the buffer selection pointer is incremented to point to the next transmit buffer. If no match is found, the buffer selection pointer continues to point to the same transmit buffer until a match is found.

Configuring Transmit Interrupts

Each Transmit Buffer[n] is associated with an Interrupt Status Bit `TXOK[n]`. This bit indicates that the message in the Transmit Buffer[n] has been successfully transmitted. The Transmit Buffers in the FlexRay controller have a dedicated interrupt line (`Flex_Txbirpt`) that is asserted when messages are successfully transmitted.

To Configure Transmit Buffers

1. Identify the Transmit Buffers that require an interrupt to be asserted on successful transmission of message.
2. Program the `TXOKE[n]` bit corresponding to the `Transmit Buffer[n]` with a '1.'

Configuring for Static Frames

Configuring a transmit buffer for transmission during the static segment involves the following steps.

1. Write the Frame ID (slot number) to the `TFID` field in the `TXBH2R[n]`.

   The `TFID` field of the `TXBH2R[n]` can be written to only when the node is in the Config state. The static frame indicated by the Transmit buffer is transmitted when the current slot number in the communication cycle is equal to the value in the TFID field. The `TFID` field can only be changed when the node is in the Config state of operation. Dynamic reconfiguration is not possible.

2. Write the computed Header `CRC` value to the THCRC field in the `TXBH2R[n]`. For static frames, the `THCRC` field once programmed in the Config state should not be updated during the Normal Active or Normal Passive states of operation.

   To compute the Header `CRC` value, use the following values:

   '0' for the Sync Frame Indicator.

   '0' for the Startup Frame Indicator.

   Frame ID as indicated by the `TFID` field in the `TXBH2R[n]`.

   Payload Length value as indicated by the `TPLEN` field in the `TXBH2R[n]`.

   For Static Frames, the `THCRC` field once computed, does not change during run-time.

3. For Static Frames, the length of the payload transmitted on the bus is always constant and indicated by the `GPLSTATIC` field in the SSPLR.
4. When the payload data stored is less than the value indicated by the `GPLSTATIC` field, the controller pads the rest of the payload bytes with 0s. The `TPLEN` field in the `TXBH2R[n]` should be programmed with the actual size of the payload data.
5. Program the `TSS` bit in the `TXBH2R[n]` to '1' if the message in the Transmit Buffer is to be transmitted only once. Program the `TSS` bit to '0' if the current message in the Transmit Buffer is to be transmitted every communication cycle.

   The `TSS` bit can be written to only when the node is in the Config state of operation.

6. To transmit a Network Management Vector, set the TNMI bit in the TXBH2R[n] to '1.' For static frames, the TNMI bit can be written to in the Normal Active or Normal Passive states only when the TBNA bit in TXBH1R[n] is a '0'.
7. The starting location of the payload (in the payload buffer) associated with the transmit buffer should be written to the TPADDR field in the TXBH3R[n]. The address should be written into TPADDR as : 0x00001000 + Payload address offset. For static frames, the TPADDR field can be written to in the Normal Active or Normal Passive states only when the TBNA bit in TXBH1R[n] is a '0.'
8. When the TBNA bit for a Transmit Buffer is '1,' the locations in the Payload Buffer that have a start address indicated by the TPADDR field in TXBH3R[n] and end address indicated by TPADDR field in TXBH3R[n] + Max Payload Size should not be written to.

Configuring for Dynamic Frames

Configuring a transmit buffer for transmission during the dynamic segment involves the following steps.

1. Write the Frame ID (slot number) to the TFID field in the TXBH2R[n].

   The TFID field of the TXBH2R can be written to only when the node is in the Config state of operation.
2. Write the computed Header CRC value to the THCRC field in the TXBH2R[n]. For dynamic frames, the THCRC field can be updated during the Normal Active or Normal Passive state of operation only TBNA bit in TXBH1R[n] is a '0'.
3. Write the TPLEN field in the TXBH2R[n] to indicate the number of 2-byte words in the payload of the message to be transmitted.

   For dynamic frames, the TPLEN can be updated during the Normal Active or Normal Passive state of operation only when the TBNA bit in TXBH1R[n] is a '0.' The value written to the TPLEN field should always be less than or equal to Max Payload Size. The value written to the TPLEN field must also be less than or equal to the protocol parameter pPayloadLengthDynMax.
4. Program the TSS bit in the TXBH2R[n] should be programmed to '1' if the message in the Transmit Buffer is to be transmitted only once. Program the TSS bit to '0', if the current message in the Transmit Buffer is to be transmitted every communication cycle.

   The TSS bit can be written to only when the node is in the Config state of operation.
5. To transmit a Message ID, set the TNMI bit in the TXBH2R[n] to '1'.

   The TNMI can be updated during the Normal Active or Normal Passive states of operation only TBNA bit in TXBH1R[n] is a '0'.
6. The starting location of the payload (in the payload buffer) associated with the transmit buffer should be written to the stored in TPADDR field in the TXBH3R[n].

   The TPADDR field can be updated during the Normal Active or Normal Passive state of operation only TBNA bit in TXBH1R[n] is a '0'.
7. When the TBNA bit for a Transmit Buffer is '1,' the locations in the Payload Buffer that are associated with the Transmit Buffer and with a starting address indicated by the TPADDR field in TXBH3R[n] and end address indicated by TPADDR field in TXBH3R[n] + Max Payload Size should not be written to.

Configuring for Startup and Sync Frames

Configuring a Transmit buffer for transmitting a Startup/ Sync Frame involves the following steps.

1. Write the Frame ID (slot number) to the TFID field in the TXBH2R[n].

The TFID field of the TXBH2R can be written to only when the node is in the Config state of operation. The Frame ID written to the TFID field must be the same as the Frame ID written to the SYSR.

2. Write the computed Header CRC value to the THCRC field in the TXBH2R[n].

   The THCRC field once programmed in the Config state should not be updated during the Normal Active or Normal Passive states.

3. Program the TPLEN field in the TXBH2R[n] with the value indicated by the SSPLR.

   The TPLEN field once programmed in the Config state should not be updated during the Normal Active or Normal Passive states.

4. Program the TSS bit in the TXBH2R[n] to '1'.

   The TSS bit can be written to only when the node is in the Config state.

5. Set the TNMI bit in the TXBH2R[n] to '0'.

   The TNMI bit once programmed in the Config state should not be updated during the Normal Active or Normal Passive states.

6. The starting location of the payload (in the payload buffer) associated with the transmit buffer should be written to the stored in TPADDR field in the TXBH3R[n].

   The TPADDR field once programmed in the Config state should not be updated during the Normal Active or Normal Passive states.

7. Write a '1' to the PKSYNC bit in the SYFCR and a '0' to the PKSUP bit in the STFCR to transmit a Sync Frame.

   The SYFCR and STFCR can be written to only in the Config state of operation.

8. Write a '1' to the PKSUP bit in the STFCR and to the PKSYNC bit in the SYFCR to transmit a Startup Frame.

   The SYFCR and STFCR can be written to only in the Config state.

Transmitting Messages

This section describes transmitting messages during Normal Active and Normal Passive operational states.

Static Frames

For Transmit Buffers that are configured to transmit in the Static Segment the following bits/bit fields can be updated during Normal Active, Normal Passive state.

1. The TBNA bit in the TXBH1R[n]. A '1' can be written to the TBNA bit to lock the Transmit Buffer for transmission.
2. The TNMI bit in the TXBH2R[n]. A '1' can be written to the TNMI bit to indicate the presence of a Network Management Vector.
3. The TPADDR field in the TXBH3R[n]. The TPADDR field can be updated to indicate the location of the Payload Data that needs to be transmitted.

The operational sequence is as follows:

1. Writing a '1' to the TBNA bit locks the transmit buffer for transmission.
2. Writing a '0' to the TBNA bit clears the TBNA bit to a '0,' provided that the message in the buffer is not currently being transmitted. If the message is being transmitted, the TBNA bit remains a '1.'
3. When the Transmit Buffer is locked, writes to the TXBH1R[n], TXBH2R[n], TXBH3R[n] are prohibited. In addition writes to Payload Buffer storage area associated with the Transmit Buffer is

prohibited. The starting address of the Payload Buffer storage area of size Max Payload Size for Transmit Buffer[n] is indicated by the TPADDR bit field.

4. When the FlexRay controller successfully transmits the message in the Transmit Buffer[n], it sets the TBNA bit in TXBH1R[n] to '0' (provided the TSS bit in the TXBH2R[n] is '1') and sets the TXOK[n] bit to a '1'. In case the TSS bit in the TXBH2R[n] is a '0', the TBNA bit in the TΞBH1P[v] continues to remain a '1', but the TXOK[n] bit will be set to a '1'.
5. The host can either poll the TXOK[n] bit or wait for an interrupt to be generated.
6. After a successful frame transmission, the host can proceed to update the transmit buffer for transmission in a successive communication cycle.

   The host can choose to update the Payload Buffer section currently associated with the Transmit Buffer or can choose to allocate a new section in the Payload Buffer that has previously been updated (by changing the TPADDR field to point to the new section in the Payload Buffer).
7. After the Transmit Buffer[n] has been updated, the TXOKC[n] bit should be written to clear the TXOK[n] bit , and a '1' should be written to the TBNA bit in the TXBH1R[n] to lock the Transmit Buffer[n] for transmission during a successive communication cycle.
8. For the Transmit Buffer to be evaluated for transmission, the TBNA should be set to '1' at least 1 slot before the scheduled transmission slot.
9. If the TBNA bit is '0' for a Transmit Buffer, a null frame is automatically transmitted during the scheduled transmission slot.

Dynamic Frames

For Transmit Buffers that are configured to transmit in the Dynamic Segment, the following bits/bit fields can be updated during Normal Active and Normal Passive states.

1. The TBNA bit in the TXBH1R[n]. A '1' can be written to the TBNA bit to lock the Transmit Buffer for transmission.
2. The TNMI bit in the TXBH2R[n]. A '1' can be written to the TNMI bit to indicate the presence of a Message ID.
3. The TPLEN field in the TXBH2R[n]. The TPLEN field can be updated to indicate the number of 2 byte words of the payload.
4. The THCRC field in the TXBH2R[n]. The THCRC field can be updated to indicate the updated Header CRC.
5. The TPADDR field in the TXBH3R[n]. The TPADDR field can be updated to indicate the location of the payload data for transmission.

The operational sequence is as follows:

1. Writing a '1' to the TBNA bit locks the transmit buffer for transmission.
2. Writing a '0' to the TBNA bit clears the TBNA bit to a '0', provided that the message in the buffer is not currently being transmitted. If the message is being transmitted, the TBNA bit remains a '1'.
3. When the Transmit Buffer is locked, writes to the TXBH1R[n], TXBH2R[n], TXBH3R[n] are prohibited.

   In addition, writes to Payload Buffer storage area associated with the Transmit Buffer is prohibited. The starting address of the Payload Buffer storage area of size Max Payload Size for Transmit Buffer[n] is indicated by the TPADDR bit field.
4. When the FlexRay controller successfully transmits the message in the Transmit Buffer[n], it sets the TBNA bit in TXBH1R[n] to '0' (provided the TSS bit in the TXBH2R[n] is '1') and sets the TXOK[n] bit to a '1'.

If the TSS bit in the TXBH2R[n] is a '0', the TBNA bit in the TXBH1R[n] continues to remain a '1', but the TXOK[n] bit will be set to a '1'.

5. The host can either poll the TXOK[n] bit or wait for an interrupt to be generated.
6. After a successful frame transmission, the host can proceed to update the Transmit Buffer for transmission in a successive communication cycle.

   The host can choose to update the payload buffer section currently associated with the transmit buffer, or to allocate a new section in the payload buffer that has previously been updated (by changing the TPADDR field to point to the new section in the storage buffer).
7. After the Transmit Buffer[n] has been updated, the TXOKC[n] bit should be written to clear the TXOK[n] bit and a '1' should be written to the TBNA bit in the TXBH1R[n] to lock the Transmit Buffer[n] for transmission during a successive communication cycle.
8. For the transmit buffer to be evaluated for transmission, the TBNA should be set to '1' at least one slot before the scheduled transmission slot.
9. If the TBNA bit is '0' for a transmit buffer, the frame transmission does not take place during the scheduled transmission slot.

Transmitting Symbols in the Symbol Window

In cluster configurations with a symbol window, writing a '1' to the SSM bit in the MTR enables transmission of MTS symbols. When the SSM bit is '1', after the transmission of a symbol in the symbol window, the SSM bit is cleared to a '0' by the controller.

Configuring Receive Buffers/ Receive FIFO

The FlexRay controller provides two types of received message buffering schemes.

- Messages that can be overwritten, are not queued, and are stored in Rx Buffer.
- Messages that cannot be overwritten, are queued behind older messages, and are stored in the Rx FIFO.

Each FlexRay Receive Buffer consists of two 32-bit words of header data and a variable number of payload words.

The following configuration guidelines are provided for the Rx Buffers and Rx FIFO.

- Every Receive Buffer[n] is associated with an acceptance filter RBAF[n]. A received message is stored in the Receive buffer only when the received message meets the filter criterion specified by the acceptance filter.
- The Receive FIFO is associated with 4 Receive FIFO acceptance filters RFAF[n]. A received message is stored in the Receive FIFO when the received message meets the filter criterion specified by any of the Receive FIFO acceptance filters.
- An Rx Buffer[n] stores received frames only if the RBAF[n] is enabled (FEN bit in RBAF2R[n] is '1').
- For Rx Buffers configured to store frames received during the static segment, the MIDE bit in the RBAF2R[n] should be '0'.
- For Rx Buffers configured to store frames received during the dynamic segment, the MIDE bit in the RBAF2R[n] may be configured either to a '0' or a '1,' depending upon whether Message ID filtering is enabled. If MIDE is '0,' the Message ID field is disregarded in the filtering process.
- The filtering engine always starts with Rx Buffer[1] and proceeds to the last configured Rx Buffer.

- If the filtering engine cannot find a valid Rx Buffer in which to store the received message, it evaluates the Receive FIFO Acceptance filters for storage in the Receive FIFO.
- If the received message matches the filtering criterion of one or more Rx Buffers, it is stored in the Rx Buffer with the lowest buffer number.
- If the received message matches the filtering criterion of a Rx Buffer and the Rx FIFO, the received message is stored in the Rx Buffer.

Configuring Receive Filters

Rx Buffers

Each Rx buffer is associated with an acceptance filter that screens incoming messages to either store or reject. Each acceptance filter (RBAF[n]) contains fields for the following parameters:

- Frame ID (FID bit field in the RBAF1R[n])
- Cycle Counter (CC bit field in the RBAF1R[n])
- Cycle Counter Mask (CCM bit field in the RBAF1R[n])
- Message ID (MID bit field in the RBAF2R[n])

Each filter also contains 3 bits (CCE and FIDE bits in RBAF1R[n] and MIDE bit in the RBAF2R[n]) to enable filtering, and one additional bit (FEN in RBAF1R[n]) to enable or disable the filter.

The filtering engine works as follows:

1. If the FEN bit in the RBAF1R[n] is a '0', no messages are stored in the Rx Buffer[n].
2. If the FEN bit in the RBAF1R[n] is a '1', the RBAF[n] is used for filtering and Rx Buffer[n] is evaluated for message storage.
3. If the FEN bit in the RBAF1R[n] is a '1', the filtering mechanism proceeds as follows:
   - If the CCE mask bit in the RBAF1R[n] is a '1', the CC bit field that is masked with the CCM bit field is compared with the cycle count of the received frame that is also masked with the CCM bit field. If there is a match, the filter field comparison is treated as a 'pass'; otherwise, the filter field comparison is treated a fail. If the CCE mask bit in the RBAF1R[n] is a '0', the CC field and the CCM field are not used for frame filtering, and the filter field comparison is treated as a pass.
   - If the FIDE mask bit in the RBAF1R[n] is a '1', the FID value in the RBAF1R[n] is compared against the frame ID of the received frame. If there is a match, the filter field comparison is treated as a pass. If the FID bit field and the frame ID of the received frame do not match, the filter field comparison is treated a fail. If the FIDE mask bit in the RBAF1R[n] is a '0', the FID field is not used for frame filtering and the filter field comparison is treated as a pass.
   - If the MIDE mask bit in the RBAF2R[n] is a '1,' the MID value in the RBAF2R[n] is compared against the message ID of the received frame. If there is a match, the filter field comparison is treated as a pass. If the MID bit field and the message ID of the received frame do not match, the filter field comparison is treated a fail. If the MIDE mask bit in the RBAF1R[n] is a '0', the MID field is not used for frame filtering, and the filter field comparison is treated as a pass.
4. For a received frame to be stored in Rx Buffer[n], the filter engine must register a pass condition on all filter field comparisons. If any one of the filter field comparisons yields a 'miss' result, the received frame is not stored in the Rx Buffer[n].
5. If a frame is not stored in Rx Buffer[n], the filter engine proceeds to evaluate Rx Buffer[n+1]. The filtering algorithm continues to search all the configured Receive Buffers and the Receive FIFO acceptance filters.

Rx FIFO

The Receive FIFO is associated with four acceptance filters that screen incoming messages for storage or rejection.

Each Receive FIFO acceptance filter (RFAF[n]) contains fields for the following parameters:

- Frame ID (FID bit field in the RFAFI1R[n])
- Cycle Counter (CC bit field in the RFAFI1R[n])
- Message ID (MID bit field in the RFAFI2R[n])
- Frame ID Mask (FIDM bit field in the RFARM1R[n])
- Cycle Counter Mask (CCM bit field in the RFARM1R[n])
- Message ID Mask (MIM bit field in the RFARM2R[n])

The following describes acceptance filter mechanics:

- The FID bit field masked with the FIDM bit field is compared to the frame ID of the received message also masked with the FIDM bit field. If there is a match, the filter is treated as a pass. If there is no match, the filter field comparison is treated as a 'fail.'
- The CC bit field that is masked with the CCM bit field is compared to the cycle count of the received message masked with the FIDM bit field. If there is a comparison match, the filter field comparison is treated as a pass. If there is not a match, it is treated as a fail.
- The MID bit field masked with the MIDM bit field is compared to the message ID of the received message that is masked with the MIDM bit field. The MID filtering is treated as a pass when the received message does not have a MID. If there is a match, the filter field comparison is treated as a pass. If no match, it is treated as a fail.
- For a frame received in the static segment/dynamic segment to be stored in Rx FIFO, the filter engine must register a pass condition on the FID, CC, and MID bit field comparisons. If any of the three filter field comparisons yield a miss result, the received frame is not stored in the Rx FIFO.
- If the received frame does not meet the criterion specified by the RFAF[n], the filter engine proceeds to evaluate RFAF[n+1]. The filtering algorithm then searches all the Receive FIFO Acceptance filters.
- If the received frame meets the criterion specified by any of the Receive FIFO Acceptance filters, the frame is stored in the Receive FIFO. If the criterion is not met, the frame is discarded.

Configuring Receive Interrupts

Each Receive Buffer[n] is associated with two Interrupt Status Bits: RXOK[n] and RXOFLW[n]. The RXOK[n] bit indicates that a message has been stored in Receive Buffer[n]. The RXOFLW[n] bit indicates that a valid message in the Receive Buffer[n] has been overwritten with a new message. The Receive Buffers in the controller have a dedicated interrupt line (Flex_Rxbirpt).

To Configure Receive Interrupts

1. Identify which Receive Buffers require an interrupt to be asserted on successful reception of a message.
2. Program the RXOKE[n] bit corresponding to the Receive Buffer[n] with a '1'.
3. Identify which Receive Buffers require an interrupt to be asserted when a valid message is overwritten by a new message.
4. Program the RXOFLWE[n] bit corresponding to the Receive Buffer[n] with a '1'.

Reading Messages from the Receive Buffer in Normal Active State

In the Normal Active/Normal Passive state, the RXOK[n] bit determines the buffer access criterion and the sanity of data in the Receive Buffer[n]. If RXOK[n] bit is '1', the Receive Buffer[n] contains a valid message and can be read by the host.

The host can either poll the RXOK[n] bits or wait for the Flex_Rxbirpt line to be asserted. During interrupt mode, the host should read the RXBISR[n] registers to determine which Receive Buffers contain a valid message.

To obtain data stored in the Rx Buffer[n], the host should perform reads on RXB[y]R[n] registers. After the read operations are complete, the RXOK[n] bit should be cleared to a '0' by performing a write to the RXOKC[n] bit in the RXBICR[n]. Proper operation of the controller is contingent on the host clearing the RXOK[n] bit for Receive Buffer[n] that has been read.

The RXOK[n] bit also acts as a status bit for the controller. The RXOK[n] bit when '1' indicates to the controller that the message in the Rx Buffer[n] has not been read by the host and is still valid. The RXOK[n] bit when '0' indicates to the controller that the message in the Rx Buffer[n] has been read by the host.

If a new message is received that needs to be stored into Rx Buffer[n] for which the RXOK[n] bit is a '1', the controller performs the following actions:

- It clears the RXOK[n] bit to a '0,' whether or not a host read operation is in progress.
- It sets the RXOFLW[n] bit to indicate that the Rx Buffer[n] locations are being overwritten with new data.
- It then overwrites the contents for RXB[y]R[n] registers whether or not a read operation is in progress.
- It sets the RXOK[n] bit back to '1' indicating that a new message is available in the Receive Buffer[n].
- The RXOFLW[n] bit continues to remain a '1'.

It is possible that the controller to overwrite the contents of the receive buffer while a host read operation on a Receive Buffer[n] is in progress. To validate the data that has been read, the host has to poll the RXOFLW[n] bit after each Receive Buffer[n] read operation.

Reading From the Receive FIFO During Normal Active State

The Receive FIFO asserts the Flex_Rxfirpt line when the following conditions occur:

- Receive FIFO empty
- Receive FIFO full
- Receive FIFO threshold crossed

Read operations on the Receive FIFO are prohibited when the Receive FIFO is empty. To read a message from a Receive FIFO, the number of read operations should be (2 + (Max Payload Size/4)), regardless of the actual message size.

Reading the Slot Status Vectors

The slot status information for each slot during a communication cycle is stored in the RSSR[n]. This includes slot status information for all slots—both when the controller is a transmitter, and when it is a receiver.

Note: Slot Status information for Transmit buffers can also be obtained using this procedure.

To obtain the Slot Status information for a frame stored in Receive Buffer[n]:

- Read the Slot ID field (indicated by RXFID field in RXB1R[n]) for the received frame stored in the Receive buffer[n].
- Based on the Frame ID value (read from the Receive Buffer[n]), determine the value of 'n' in RSSR[n]. (n indicates the nth RSSR[n].) Each RSSR[n] stores the slot status information for the 4n, 4n-1, 4n-2 and 4n-3 slots.
- Read the RSSR[n] and then choose the appropriate byte to obtain the slot status information.
  - RSSR[n][0..7] indicates the status information for the slot 4n-3.
  - RSSR[n][8..15] indicates the status information for the slot 4n-2.
  - RSSR[n][16..23] indicates the status information for the slot 4n-1.
  - RSSR[n][24..31] indicates the status information for the slot 4n.

Receiving Symbols During Symbol Window

Reception of signals during the symbol window is indicated by the SVALSYM status bit in the SWSR, and also by the MTSRX bit in the CISR.

Absolute and Relative Timers

Absolute Timer Operation

The Absolute Timer sets the ABTIM bit in the CISR when the programmed value of cycle count and macrotick offset is reached. It can be used to indicate the occurrence of a particular macrotick in a particular communication cycle.

The Absolute Timer sets the ABTIM bit in the CISR when the following conditions are true:

- ATEN bit is '1', AND
- ATCID bit field in the ATCCR "AND" ATCMACK bit field in the ATCCR = Current cycle count value (VCYCLECNT bit field in the CCSR) "AND" ATCMASK bit field in the ATCCR, AND
- ATMAOFF bit field in the ATCCR = MCTICK bit field in the MSR.

The "AND" operation indicates a *logical and operation*. The Absolute Timer behaves as a repetitive timer if the ATMODE bit is '1.'

Absolute Timer Configuration

To enable the timer, the ATEN bit in the ATCR should be set to '1.' To disable the timer, the ATEN bit should be set to '0.' The timer can be disabled anytime. The ATEN bit can be written to only when controller is in Normal Active or Normal Passive states.

Program the ATCID and ATCMASK fields with the appropriate cycle ID and cycle mask fields. If the cycle mask is all '0s,' the Absolute Timer is set every communication cycle when the MCTICK bit field in the MSR is equal to the ATMAOFF bit field in the ATCCR. The ATCID and ATCMASK fields in the ATCCR, and the ATMODE and ATMAOFF bit fields in the ATMOR can be programmed only when the ATEN bit in the ATCR is a '0'. Program the ATMAOFF field in the ATMOR with the appropriate macrotick offset value.

To configure the timer as repetitive, program the ATMODE bit to a '1'.

Relative Timer Operation

The relative timer sets the RELTIM bit in the CISR when the macrotick time indicated by the RTMAOFF field in the RTMOR has elapsed. The relative timer can be used to indicate a time duration indicated in macroticks.

The relative timer sets the RELTIM bit in the CISR when the following conditions are true:

- RTEN bit is '1', and
- A duration of time in macroticks (as indicated by the RTMAOFF field in the RTMOR) has elapsed since the RTEN bit was changed to a '1' from a '0'.

The relative timer behaves as a repetitive timer if the RTMODE bit is '1.'

Relative Timer Configuration

The configuration steps for the telative Timer are as follows:

To enable the timer, the RTEN bit in the RTCR should be set to '1'. To disable the timer, the RTEN bit should be set to '0'. The timer can be disabled anytime. The RTEN bit can be written to only when controller is in the Normal Active or Normal Passive state of operation.

Program the RTMAOFF field in the RTMOR with the appropriate macrotick offset value. The RTMAOFF field should never be X"00000000". To configure the timer as repetitive, program the RTMODE bit in the RTMOR to a '1'.

The RTMODE and RTMAOFF bit fields in the RTMOR can be programmed only when the RTEN bit in the RTCR is a '0'.

Design Constraints

Location Constraints

The external pins of the Xilinx FlexRay core should be connected to the corresponding pins of the FlexRay PHY.

Timing Constraints

The core has three different clock domains: PLB_Clk, Flex_Clk and Sample_Clk. Timing Ignore constraints should be added to isolate these clock domains. The constraints given below can be used with the Xilinx FlexRay Controller.

PERIOD Constraints for Clock Nets

PLB_Clk

The clock provided to PLB_Clk must be constrained to a clock frequency of 50 MHz - 100 MHz.

Flex_Clk

The Flex_Clk is generated using DCMs and has a fixed frequency of 40MHz.

```
NET "plb_flexray_0_Flex_Clk" TNM_NET = "Flex_Clk";
TIMESPEC "TS_Flex_Clk" = PERIOD "Flex_Clk" 25 ns HIGH 50%;
```

Sample_Clk

The Sample_Clk is generated using DCMs and has a fixed frequency of 80MHz.

```
NET "plb_flexray_0_Sample_Clk" TNM_NET = "SAMPLE_CLK";
TIMESPEC "TS_SAMPLE_CLK" = PERIOD "SAMPLE_CLK" 12.5 ns HIGH 50%;
```

Design Implementation

Target Technology

The Xilinx FlexRay Controller can be implemented in Spartan 3, Spartan-3 XA, Spartan 3E, Spartan-3E XA, Spartan-3A, Spartan-3AN, Virtex-II Pro, and Virtex-4 devices. The device used must have the fol-

lowing attributes:

- Large enough to accommodate the core.
- Contains a sufficient number of IOBs.

Device Utilization and Performance Benchmarks

Below is an example FPGA performance and resource utilization benchmarks table for Spartan-3 (xc3s1500fg456-4).

Table 231: **XPS FlexRay FPGA Performance and Resource Utilization Benchmarks**

Parameter Values				Device Resources			f_{MAX} (MHz)
C_MAX_PAYLOAD	C_NUM_TX_BUF	C_NUM_RX_BUF	C_RX_FIFO_DPTH	Slices	Slice Flip-Flops	4-input LUTs	PLB f_{MAX}
4	2	2	2	4153	3380	5079	92.429
4	4	4	4	4211	3397	5133	97.837
4	8	8	8	4301	3426	5215	93.196
4	16	16	16	4425	3466	5356	88.589
4	32	32	32	4768	3568	5651	91.717
4	64	64	64	4788	3573	5675	70.891
4	128	128	128	4794	3578	5676	77.166
256	2	2	2	4156	3380	5085	92.635
256	4	4	4	4213	3397	5139	100.664
256	8	8	8	4333	3431	5264	93.005
256	16	16	16	4533	3476	5499	84.947
256	32	32	32	4894	3578	5807	76.793
256	64	64	64	4906	3582	5834	76.869

Specification Exceptions

N/A

Reference Documents

1 *FlexRay Communication System Protocol Specification Version 2.1 Revision A*

2. *IBM 128-bit Processor Local Bus Architecture Specifications version 4.6*

Revision History

Date	Version	Revision
10/30/2007	1.0	Initial Xilinx release

09/29/09 - This is the final publication. No content was changed.