MFR4310

Reference Manual

FlexRay Communication Controllers

MFR4310RM
Rev. 2
03/2008

freescale.com

freescale™ semiconductor

MFR4310 Reference Manual

MFR4310RM
Rev. 2
03/2008

freescale™
semiconductor

To provide the most up-to-date information, the revision of our documents on the World Wide Web will be the most current. Your printed copy may be an earlier revision. To verify that you have the latest information available, refer to http://www.freescale.com/flexray.

The following revision history table summarizes changes made to this document.

Revision History

Date	Revision Level	Description
9 May 2007	0	First public release.
20 Jun 2007	1	Added row for 1M63J maskset to Table 2-2. Changed Figure 4-2 read and reset values and following paragraph to reflect 1M63J maskset as an example.
21 Mar 2008	2	Revised Figure 1-1. Updated Table A-1 (maximum junction temperature changed from +150C to +140C). Updated Table A-5. Thermal Characteristics Updated Table A-8. Supply Current Characteristics (50mA max for -40 C, 25C & 140 C). Updated Table A-12. Oscillator Characteristics (VDCbias TYP = 2.5).

Contents

Section Number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Device Overview

Chapter 3
FlexRay Module (FLEXRAYV4)

Chapter 4
Port Integration Module (PIM)

Chapter 5
Dual Output Voltage Regulator (VREG3V3V2)

Chapter 6
Clocks and Reset Generator (CRG)

Chapter 7
Oscillator (OSCV2)

Appendix A
Electrical Characteristics

Appendix B
Package Information

Appendix C
Printed Circuit Board Layout Recommendations

Appendix D
Index of Registers

Section Number	Title	Page

List of Figures

Figure Number	Title	Page

Figure Number	Title	Page

Figure Number	Title	Page

Figure Number Title Page

Figure Number	Title	Page

List of Tables

Table Number **Title** **Page**

Table Number	Title	Page

Table Number	Title	Page

Table Number	Title	Page

Table Number	Title	Page

Table Number	Title	Page

Chapter 1
Introduction

This reference manual provides information on a system that includes the MFR4310 FlexRay Communication Controller Module.

1.1 Audience

This reference manual is intended for application and system hardware developers who wish to develop products for the FlexRay MFR4310. It is assumed that the reader understands FlexRay protocol functionality and microcontroller system design.

1.2 Additional Reading

For additional reading that provides background to, or supplements, the information in this manual:

- For more information about the FlexRayprotocol, refer to the following document:
  - *FlexRay Communications System Protocol Specification V2.1A*
  - *FlexRay Communications System Electrical Physical Layer Specification V2.1A*
- For more information about M9HCS12, MPC5xx and MPC55xx Familydevices and how to program them, refer to the Freescale Products section at www.freescale.com.

1.3 Terminology

Table 1-1. Acronyms and Abbreviations

Term	Meaning
AMI	Asynchronous Memory Interface
BCU	Buffer Control Unit
CC	Communication Controller
CDC	Clock Domain Crosser
CHI	Controller Host Interface
ID	Identification
EBI	External Bus Interface
FRM	FlexRay Memory
FSS	Frame Start Sequence
HCS12	Freescale's HCS12 family of microcontrollers
HIF	Host Interface
LUT	Look Up Table
MBIDX	Message Buffer Index
MBNum	Message Buffer Number
MCU	Microcontroller Unit
MPC	Device title prefix for Freescale's MPC5xx and MPC55xx family microcontrollers
μT	Microtick. A microtick is one CLK_CC period long, and starts on the rising edge of CLK_CC.
MT	Macrotick
MTS	Media Access Test Symbol
NIT	Network Idle Time
PE	Protocol Engine
PHY	Physical Layer Interface
PL	Physical Layer
POC	Protocol Operation Control
SEQ	Sequencer Engine
Rx	Reception
TCU	Time Control Unit
Tx	Transmission

Table 1-2. Notational Conventions

active-high	Names of signals that are active-high are shown in upper case text, without a # symbol at the end. Active-high signals are asserted (active) when they are high and deasserted when they are low.
active-low	An active-low signal is asserted (active) when it is at the logic low level and is deasserted when it is at the logic high level. **Note:** A # symbol at the end of a signal name indicates that the signal is active-low.
asserted	A signal that is asserted is in its active logic state. An active-low signal changes from high to low when asserted; an active-high signal changes from low to high when asserted.
deasserted	A signal that is deasserted is in its inactive logic state. An active-low signal changes from low to high when deasserted; an active-high signal changes from high to low when deasserted.
set	To set a bit means to establish logic level one on the bit.
clear	To clear a bit means to establish logic level zero on the bit.
0x0F	The prefix 0x denotes a hexadecimal number.
0b0011	The prefix 0b denotes a binary number.
x	In certain contexts, such as a signal encoding, this indicates don't care. For example, if a field is binary encoded 0bx001, the state of the most significant bit is don't care.
==	Used in equations, this symbol signifies comparison.
#	A # symbol at the end of a signal name indicates that the signal is active-low.

1.4 Part Number Coding

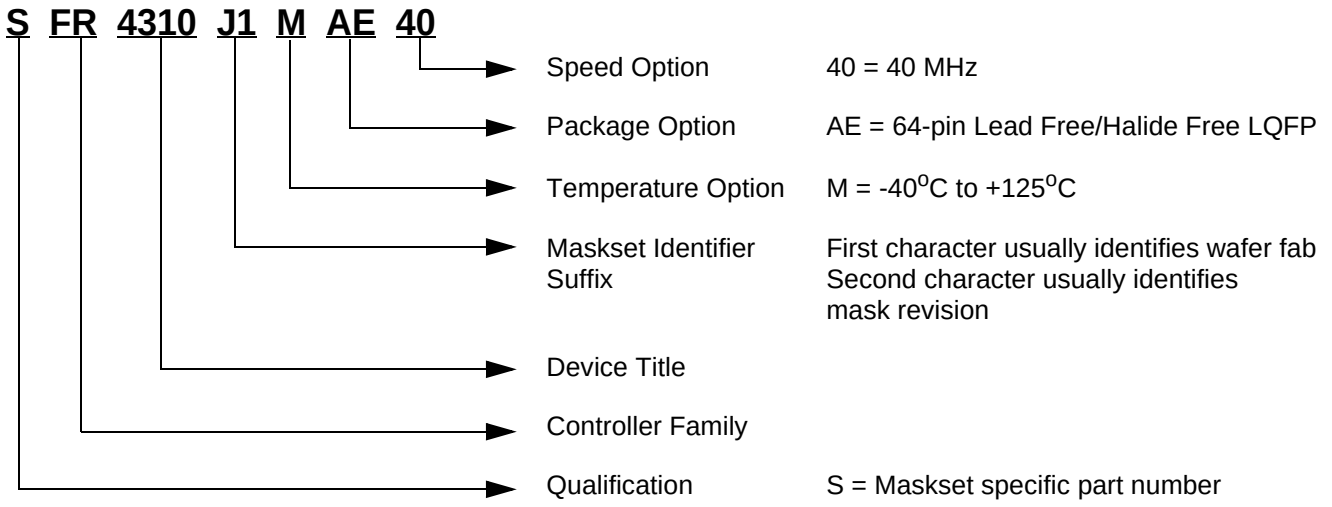


Figure 1-1. Order Part Number Coding

Chapter 2
Device Overview

2.1 Introduction

The MFR4310 FlexRay Communication Controller implements the FlexRay protocol according to the *FlexRay Communications System Protocol Specification V2.1A.*

The controller host interface (CHI) of the MFR4310 FlexRay Communication Controller is implemented in accordance with Chapter 3, "FlexRay Module (FLEXRAYV4)" of this reference manual.

2.2 Features

The MFR4310 FlexRay controller provides the following features:

- Single channel support
  - Internal channel A and FlexRay Port A can be configured to be connected to physical FlexRay channel A or physical FlexRay channel B
- Variable bit rate support: 2.5, 5, 8, or 10 Mb/s
- 128 configurable message buffers with
  - Individual frame ID filtering
  - Individual channel ID filtering
  - Individual cycle counter filtering
- Message buffer header, status and payload data are stored in FlexRay memory
  - Consistent data access ensured by means of buffer locking scheme
  - Host can lock multiple buffers at the same time
- Size of message buffer data section configurable from 0 up to 254 bytes
- Two independent message buffer segments with configurable size of payload data section
  - Each segment can contain message buffers assigned to the static segment and message buffers assigned to the dynamic segment at the same time
- Zero padding for transmit message buffers in static segment
  - Applied when the frame payload length exceeds the size of the message buffer data section
- Transmit message buffers configurable with state/event semantics
- Message buffers can be configured as
  - Receive message buffers
  - Single buffered transmit message buffer
  - Double buffered transmit message buffer (combines two single buffered message buffer)
- Individual message buffer reconfiguration supported
  - Means provided to safely disable individual message buffers
  - Disabled message buffers can be reconfigured

- Two independent receive FIFOs
  - One receive FIFO per channel
  - Up to 256 entries for each FIFO
  - Global frame ID filtering, based on both value/mask filters and range filters
  - Global channel ID filtering
  - Global message ID filtering for the dynamic segment
- Four configurable slot error counters
- Four dedicated slot status indicators
  - Used to observe slots without using receive message buffers
- Provides measured value indicators for clock synchronization
  - PE internal synchronization frame ID and measurement tables can be copied into the FlexRay memory
- Fractional macroticks are supported for clock correction
- Maskable interrupt sources provided through individual and combined interrupt lines
- One absolute timer
- One timer that can be configured to absolute or relative

Features specific to the MFR4310 include the following:

- Identical pinout to MFR4300; pin functionality compatible with MFR4300
- Three hardware selectable host interfaces:
  - HCS12 Interface for direct connection to Freescale's HCS12 family of microcontrollers, with interface clock signal to synchronize the data transfer (the maximum frequency of this clock signal can be calculated from the ECLK_CC pulse width low and high times, t_{LEC} and t_{HEC} given in Table A-15.)
  - Asynchronous Memory Interface (AMI) for asynchronous connection to microcontrollers — minimum read access time of 56 ns (with CHICLK_CC running at 76 MHz)
  - MPC Interface for asynchronous connection to Freescale's MPC5xx and MPC55xx family microcontrollers — minimum read access time of 56 ns (with CHICLK_CC running at 76 MHz)
- 8K bytes addressable for byte or word accesses
- Internal quartz oscillator of 40 MHz
- CHI and AMI/MPC clock selectable between 40 MHz oscillator clock used for PE and 20 MHz to 76 MHz separate CHI/AMI/MPC-only clock
- Internal voltage regulator for the digital logic and the oscillator
- Hardware selectable clock output to drive external host devices: disabled, 4, 10, or 40 MHz
- Maskable interrupt sources available over one interrupt output line
- RESET# glitch filter
- Electrical physical layer interface compatible with dedicated FlexRay physical layer
- Four multiplexed debug strobe pins

2.3 Block Diagram

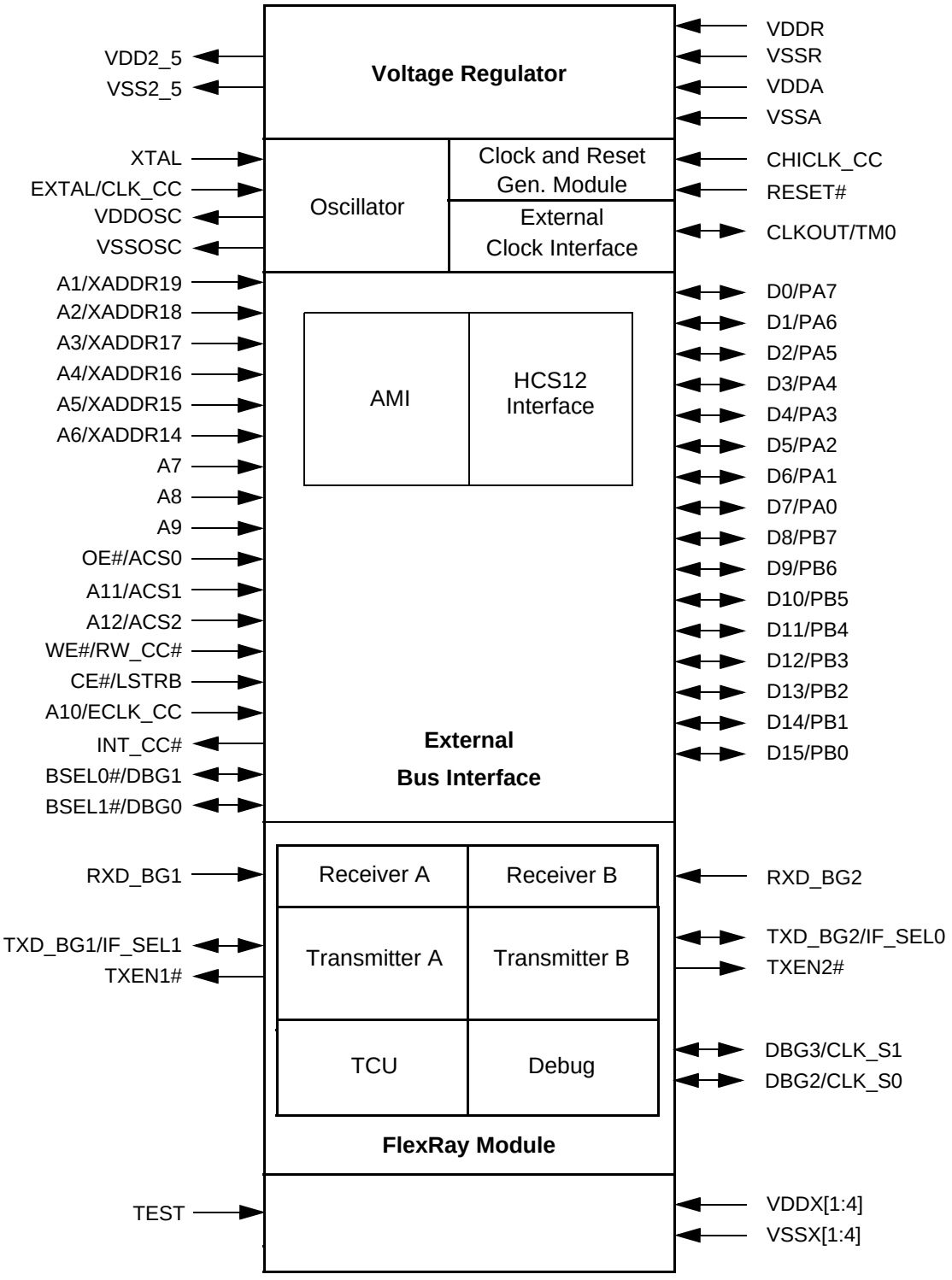


Figure 2-1. MFR4310 Functional Block Diagram

2.3.1 Memory Map

Table 2-1 shows the MFR4310 device memory map.

Table 2-1. MFR4310 Device Memory Map After Reset

address (Hex)	Module	Registers	Size (bytes)
0x0000–0x000E	FlexRay[1]	Configuration and Control Registers	16
0x0010–0x0012	FlexRay	Reserved	4
0x0014–0x0026	FlexRay	Interrupt and Error Handling Registers	20
0x0028–0x003E	FlexRay	Protocol Status Registers	24
0x0040–0x0044	FlexRay	Sync Frame Counter and Table Registers	6
0x0046–0x004A	FlexRay	Sync Frame Filter Registers	6
0x004C–0x0058	FlexRay	Network Management Vector Registers	14
0x005A–0x0062	FlexRay	Timer Configuration Registers	10
0x0064–0x0066	FlexRay	Slot Status Configuration Registers	4
0x0068–0x007E	FlexRay	Slot Status and Slot Status Counter Registers	24
0x0080–0x0082	FlexRay	MTS Generation Registers	4
0x0084	FlexRay	Shadow Buffer Configuration Register	2
0x0086–0x008A	FlexRay	Receive FIFO — Configuration	6
0x008C–0x008E	FlexRay	Receive FIFO — Status	4
0x0090–0x009A	FlexRay	Receive FIFO — Filter	12
0x009C, 0x009E	FlexRay	Dynamic Segment Status Registers	4
0x00A0–0x00DE	FlexRay	Protocol Configuration Registers	64
0x00E0–0x00E2	CRG[2]	Clock and Reset Generation Registers	4
0x00E4–0x00EE	FlexRay	Reserved	12
0x00F0–0x00FE	PIM[3]	Part ID, ASIC Version Number, and Interface Pin Drive Strength and Pullup/pulldown Control and Enable Registers	16
0x0100–0x01FE	FlexRay	Message Buffers Configuration, Control, Status (Message Buffer 0–31)	256
0x0200–0x02FE	FlexRay	Message Buffers Configuration, Control, Status (Message Buffer 32–63)	256
0x0300–0x03FE	FlexRay	Message Buffers Configuration, Control, Status (Message Buffer 64–95)	256
0x0400–0x04FE	FlexRay	Message Buffers Configuration, Control, Status (Message Buffer 96–127)	256
0x0500–0x07FE	FlexRay	Reserved	768
0x0800–0x1FFE	FlexRay	Message Buffers and FIFO Frame Header/Offset/Status/Data	6144

[1] For detailed information on the MFR4310 FlexRay module registers, see Chapter 3, “FlexRay Module (FLEXRAYV4).

[2] For detailed information on the MFR4310 CRG module registers, see Chapter 6, “Clocks and Reset Generator (CRG)”.

[3] For detailed information on the MFR4310 PIM module registers, see Chapter 4, “Port Integration Module (PIM)”.

2.3.2 Part ID and Module Version Number Assignments

Three 16-bit read-only registers provide information about the device and the MFR4310 FlexRay module (see Table 2-2).

Table 2-2. Part ID and Module Version Numbers

Device	Mask Set Number	Part ID		
		PIDR	AVNR	MVR
MFR4310	0M63J	4310	0000	8566
MFR4310	1M63J	4310	0001	8566

The PIDR (see Section 4.3.1.1, “Part ID Register (PIDR)”) provides the part ID number in binary coded decimal.

The AVNR (see Section 4.3.1.2, “ASIC Version Number Register (AVNR)”) provides the ASIC version number in binary coded decimal.

The MVR (see Section 3.3.2.3, “Module Version Register (MVR)”) provides the FlexRay module version number in binary coded decimal. Bits 15 to 8 of the MVR comprise the controller host interface (CHI) version number; bits 7 to 0 comprise the protocol engine (PE) version number.

These read-only values provide a unique ID for each revision of the device.

2.4 Signal Descriptions

2.4.1 System Pinout

The MFR4310 is available in a 64-pin low profile quad flat package (LQFP). Most pins perform two functions, as described in Section 2.4.2, “Pin Functions and Signal Properties”. Figure 2-2 shows the pin assignments.

NOTE

For a recommended printed circuit board layout, see Appendix C, “Printed Circuit Board Layout Recommendations”.

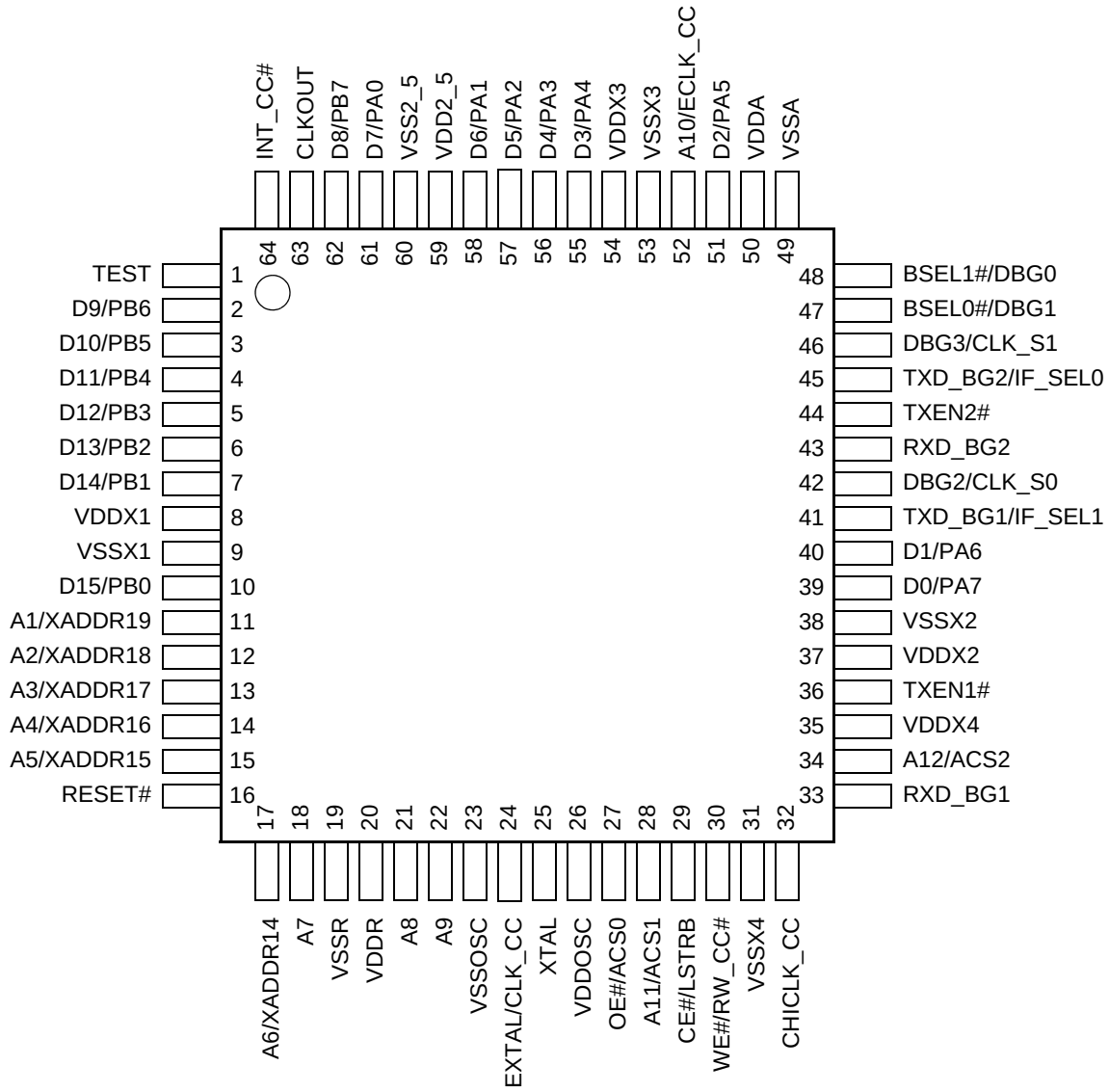


Figure 2-2. MFR4310 Pin Assignment

2.4.2 Pin Functions and Signal Properties

Table 2-3. Pin Functions and Signal Properties

Pin #	Pin Name[1]		Powered by	I/O	Pin Type[2, 3]	Reset	Functional Description
	Function 1	Function 2					
Host Interface Pins							
11	A1	XADDR19	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines. A1 is the LSB of the AMI/MPC address bus; XADDR14 is the LSB of the HCS12 expanded address lines
12	A2	XADDR18	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines.
13	A3	XADDR17	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines.
14	A4	XADDR16	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines.
15	A5	XADDR15	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines.
17	A6	XADDR14	VDDX	I	PC	-	AMI/MPC address bus; HCS12 expanded address lines.
18	A7	-	VDDX	I	PC	-	AMI/MPC address bus
21	A8	-	VDDX	I	PC	-	AMI/MPC address bus
22	A9	-	VDDX	I	PC	-	AMI/MPC address bus
27	OE#	ACS0	VDDX	I	PC	-	AMI/MPC read output enable signal; HCS12 address select input
28	A11	ACS1	VDDX	I	PC	-	AMI/MPC address bus; HCS12 address select inputs
34	A12	ACS2	VDDX	I	PC	-	AMI/MPC address bus; HCS12 address select inputs
48	BSEL1#	DBG0	VDDX	I/O	PC	-	AMI/MPC byte select; Debug strobe point
47	BSEL0#	DBG1	VDDX	I/O	PC	-	AMI/MPC byte select; Debug strobe point
10	D15	PB0	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus. D15 is the MSB of the AMI/MPC data bus; PB0 is the LSB of the HCS12 address/data bus
7	D14	PB1	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
6	D13	PB2	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus

Table 2-3. Pin Functions and Signal Properties (continued)

Pin #	Pin Name[1]		Powered by	I/O	Pin Type[2, 3]	Reset	Functional Description
	Function 1	Function 2					
5	D12	PB3	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
4	D11	PB4	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
3	D10	PB5	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
2	D9	PB6	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
62	D8	PB7	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
61	D7	PA0	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
58	D6	PA1	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
57	D5	PA2	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
56	D4	PA3	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
55	D3	PA4	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
51	D2	PA5	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
40	D1	PA6	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus
39	D0	PA7	VDDX	I/O	Z/DC/PC	Z	AMI/MPC data bus; HCS12 multiplexed address/data bus. D0 is the LSB of the AMI data bus; PA7 is the MSB of the HCS12 address/data bus
29	CE#	LSTRB	VDDX	I	PC	-	AMI/MPC chip select signal; HCS12 low-byte strobe signal
30	WE#	RW_CC#	VDDX	I	PC	-	AMI write enable signal; HCS12 read/write select signal
52	A10	ECLK_CC	VDDX	I	PC	-	AMI/MPC address bus; HCS12 clock input
Physical Layer Interface							
33	RXD_BG1	-	VDDX	I	PC	-	PHY Data receiver input
43	RXD_BG2	-	VDDX	I	PC	-	PHY Data receiver input
36	TXEN1#	-	VDDX	O	DC	1	Transmit enable for PHY

Table 2-3. Pin Functions and Signal Properties (continued)

Pin #	Pin Name[1]		Powered by	I/O	Pin Type[2, 3]	Reset	Functional Description
	Function 1	Function 2					
44	TXEN2#	-	VDDX	O	DC	1	Transmit enable for PHY
45	TXD_BG2	IF_SEL0	VDDX	I/O	DC/PU	-	PHY Data transmitter output / Host interface select
41	TXD_BG1	IF_SEL1	VDDX	I/O	DC/PD	-	PHY Data transmitter output / Host interface select
Clock Signals							
32	CHICLK_CC	-	VDDX	I	-	-	External CHI clock input – *selectable*
63	CLKOUT	-	VDDX	I/O	DC	-	Controller clock output – *selectable as disabled/4/10/40 MHz*
Others							
16	RESET#	-	VDDX	I	PD	-	External hardware reset input
64	INT_CC#	-	VDDX	O	OD/DC	0	Controller level-sensitive interrupt output
1	TEST	-	VDDX	I	PD	-	Factory Test mode select – *must be tied to logic low in application*
42	DBG2	CLK_S0	VDDX	I/O	DC/PD	-	Debug strobe point / Output clock select
46	DBG3	CLK_S1	VDDX	I/O	DC/PD	-	Debug strobe point / Output clock select
Oscillator							
24	EXTAL	CLK_CC	VDDOSC	I	-	-	Crystal driver / External clock
25	XTAL	-	-	I	-	-	Crystal driver
Supply/Bypass Filter pins							
8	VDDX1	-	-	-	-	-	Supply voltage, I/O
37	VDDX2	-	-	-	-	-	Supply voltage, I/O
54	VDDX3	-	-	-	-	-	Supply voltage, I/O
35	VDDX4	-	-	-	-	-	Supply voltage, I/O
9	VSSX1	-	-	-	-	-	Supply voltage ground, I/O
38	VSSX2	-	-	-	-	-	Supply voltage ground, I/O
53	VSSX3	-	-	-	-	-	Supply voltage ground, I/O
31	VSSX4	-	-	-	-	-	Supply voltage ground, I/O
20	VDDR	-	-	-	-	-	Supply voltage, supply to pin drivers and internal Voltage Regulator
19	VSSR	-	-	-	-	-	Supply voltage ground, ground to pin drivers and internal Voltage Regulator
50	VDDA	-	-	-	-	-	Supply analog voltage
49	VSSA	-	-	-	-	-	Supply analog voltage ground

Table 2-3. Pin Functions and Signal Properties (continued)

Pin #	Pin Name[1]		Powered by	I/O	Pin Type[2, 3]	Reset	Functional Description
	Function 1	Function 2					
59	VDD2_5[4]	-	-	-	-	-	Core voltage power supply output (nominally 2.5V)
60	VSS2_5[4]	-	-	-	-	-	Core voltage ground output
26	VDDOSC[4]	-	-	-	-	-	Oscillator voltage power supply output (nominally 2.5V)
23	VSSOSC[4]	-	-	-	-	-	Oscillator voltage ground output

1 # – signal is active-low

2 Acronyms:
PC – (Pullup/pulldown Controlled) Register controlled internal weak pullup/pulldown for a pin in the input mode. Refer to the following sections for more information:
– Section 4.3.1.5, “Host Interface Pins Pullup/pulldown Enable Register (HIPPER)”
– Section 4.3.1.6, “Host Interface Pins Pullup/pulldown Control Register (HIPPCR)”
– Section 4.3.1.7, “Physical Layer Pins Pullup/pulldown Enable Register (PLPPER)”
– Section 4.3.1.8, “Physical Layer Pins Pullup/pulldown Control Register (PLPPCR)”
PU/PD – (Pullup/Pulldown) Internal weak pullup/pulldown for a pin in the input mode
DC – (Drive strength Controlled) Register controlled drive strength for a pin in the output mode. Refer to the following sections for more information:
– Section 4.3.1.3, “Host Interface Pins Drive Strength Register (HIPDSR)”
– Section 4.3.1.4, “Physical Layer Pins Drive Strength Register (PLPDSR)”
Z – Tristated pin
OD – (Open Drain) Output pin with open drain

3 Reset state:
All pins with the PC option – pullup/pulldown is disabled,
all pins with the DC option – have full drive strength

4 No load allowed except for bypass capacitors.

2.4.3 Detailed Signal Descriptions

2.4.3.1 A[6:1]/XADDR[14:19] — AMI/MPC Address Bus; HCS12 Expanded Address Inputs

A[6:1]/XADDR[14:19] are general purpose input pins. Their function is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pins can be configured to enable or disable pullup or pulldown resistors on the pins. (See Section 4.3.1.5, “Host Interface Pins Pullup/pulldown Enable Register (HIPPER)” and Section 4.3.1.6, “Host Interface Pins Pullup/pulldown Control Register (HIPPCR)”.)

A[6:1] are AMI/MPC interface address signals. A1 is the LSB of the AMI/MPC address bus.

XADDR[14:19] are HCS12 interface expanded address lines. XADDR14 is the LSB of the HCS12 interface expanded address lines.

2.4.3.2 A[9:7] — AMI/MPC Address Bus

A[9:7] are general purpose input pins. Their function is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pins can be configured to enable or disable pullup or pulldown resistors on the pins.

A[9:7] are AMI/MPC interface address signals.

2.4.3.3 OE#/ACS0 — AMI/MPC Read Output Enable, HCS12 Address Select Input

OE#/ACS0 is a general purpose input pin. Its function is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pin can be configured to enable or disable a pullup or pulldown resistor on the pin.

OE# is the AMI/MPC interface output enable signal. This signal controls MFR4310 data output and the state of three-stated data pins D[15:0] during host read operations.

ACS0 is an HCS12 interface address select signal.

2.4.3.4 A[12:11]/ACS[2:1] — AMI/MPC Address Bus, HCS12 Expanded Address Inputs

A[12:11]/ACS[2:1] are general purpose input pins. Their function is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pins can be configured to enable or disable pullup or pulldown resistors on the pins.

A[12:11] are AMI/MPC interface address signals.

ACS[1:2] are HCS12 interface address select signals.

2.4.3.5 BSEL[1:0]#/DBG[0:1] — AMI/MPC Byte Select, Debug Strobe Points

BSEL[1:0]#/DBG[0:1] are general purpose input or output pins. Their function is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pins can be configured to provide high or reduced output drive, and also to enable or disable pullup or pulldown resistors on the pins.

BSEL[1:0]# are AMI/MPC byte select signals.

DBG[0:1] are debug strobe point output signals. The functions output on these pins are selected by the debug port control register. Refer to Section 3.4.16, “Strobe Signal Support” for more information.

2.4.3.6 D[15:8]/PB[0:7] — AMI/MPC Data Bus, HCS12 Multiplexed Address/Data Bus

D[15:8]/PB[0:7] are general purpose input or output pins. Their functions are selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. These pins can be configured to provide high or reduced output drive, and also to enable or disable pullup or pulldown resistors on the pins.

D[15:8] are data signals of the AMI/MPC interface. D15 is the MSB of the AMI/MPC data bus.

PB[0:7] are HCS12 interface multiplexed address/data signals in the HCS12 Host interface mode of operation. PB0 is the LSB of the HCS12 address/data bus.

2.4.3.7 D[7:0]/PA[0:7] — AMI/MPC Data Bus, HCS12 Multiplexed Address/Data Bus

D[7:0]/PA[0:7] are general purpose input or output pins. Their functions are selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. These pins can be configured to provide high or reduced output drive, and also to enable or disable pullup or pulldown resistors on the pins.

D[7:0] are data signals of the AMI/MPC interface. D0 is the LSB of the AMI/MPC data bus.

PA[0:7] are HCS12 interface multiplexed address/data signals in the HCS12 Host interface mode of operation. PA7 is the MSB of the HCS12 address/data bus.

2.4.3.8 CE#/LSTRB — AMI/MPC Chip Select, HCS12 Low-byte Strobe

The function of this pin is selected by IF_SEL[1:0] pins. Refer Section 2.7, “External Host Interface” for more information. The pin can be configured to enable or disable a pullup or pulldown resistor on the pin.

CE# is an AMI/MPC interface transfer size input signal. It indicates the size of the requested data transfer in the current bus cycle.

LSTRB is an HCS12 interface low-byte strobe input signal. It indicates the type of bus access.

2.4.3.9 WE#/RW_CC# — AMI Write Enable, HCS12 Read/Write Select

The function of this pin is selected by the IF_SEL[1:0] pins. Refer to Section 2.7, “External Host Interface” for more information. The pin can be configured to enable or disable a pullup or pulldown resistor on the pin.

WE# is an AMI interface write select signal. It strobes the valid data provided by the host on the D[15:0] pins during write operations to the MFR4310 memory.

RW_CC# is an HCS12 interface read/write input signal. It indicates the direction of data transfer for a transaction.

2.4.3.10 A10/ECLK_CC — AMI/MPC Address Bus, HCS12 Clock Input

The function of this pin is selected by the IF_SEL[1:0] pins. Refer Section 2.7, “External Host Interface” for more information. The pin can be configured to enable or disable a pullup or pulldown resistor on the pin.

A10 is an AMI/MPC interface address signal.

ECLK_CC is the HCS12 interface clock input signal. (The maximum frequency of this signal can be calculated from the ECLK_CC pulse width low and high times, t_{LEC} and t_{HEC} given in Table A-15.)

2.4.3.11 RXD_BG[2:1] — PHY Data Receiver Inputs

RXD_BG[2:1] are bus driver receive data input signals if the FlexRay Optical/Electrical PHY is configured:

- RXD_BG1 is the input to the CC from Physical Layer Channel 1
- RXD_BG2 is the input to the CC from Physical Layer Channel 2

These pins can be configured to enable or disable pullup or pulldown resistors on the pins.

2.4.3.12 TXEN[2:1]# — PHY Transmit Enable

TXEN[2:1]# are bus driver transmit enable output signals if the FlexRay Optical/Electrical PHY is configured:

- TXEN1# is the output of the CC to Physical Layer Channel 1
- TXEN2# is the output of the CC to Physical Layer Channel 2

These pins can be configured to provide high or reduced output drive.

2.4.3.13 TXD_BG[1:2]/IF_SEL[1:0] — PHY Transmit Data Outputs, Host Interface Selection

These pins can be configured to provide high or reduced output drive.

TXD_BG[1:2] are bus driver transmit data output signals if the FlexRay Optical/Electrical PHY is configured:

- TXD_BG1 is the output of the CCto Physical Layer Channel 1
- TXD_BG2 is the output of the CCto Physical Layer Channel 2

IF_SEL[1:0] are the CC external interface selection input signals. Refer to Table 2-6 for the selection coding.

NOTE

The IF_SEL[1:0] signals are inputs during the internal reset sequence and are latched during the internal reset sequence.

While the IF_SEL[1:0] levels are being latched, the output drive control is disabled and the internal pull resistors are connected (pullup on IF_SEL0; pulldown on IF_SEL1).

As IF_SEL[1:0] signals share pins with Physical Layer Interface signals, pullup/pulldown devices must be used for the selection. Recommended pullup/pulldown resistor values for the IF_SEL[1:0] inputs are given in Section 2.6.3, “Recommended Pullup/pulldown Resistor Values”.

2.4.3.14 CHICLK_CC — External CHI Clock Input

CHICLK_CC is the selectable external CHI clock input. It can be selected to drive the Asynchronous Memory Interface (see Section 2.6.2, “External Host Interface Selection”).

2.4.3.15 CLKOUT — Clock Output

CLKOUT is a continuous clock output signal. The frequency of CLKOUT is selected by the CLK_S[1:0] pins. The CLKOUT signal, if enabled, is always active:

1. after power-up of the CC,
2. after a low-voltage reset,
3. after a clock monitor failure reset,
4. during and after an external hard reset.

The pin can be configured to provide high or reduced output drive.

NOTE

As the CLKOUT signal can be disabled during internal resets, refer to Section 6.4.3, “CLKOUT Mode Selection and Control” for more information on CLKOUT generation during external hard and internal resets.

2.4.3.16 RESET# — External Reset

RESET# is an active-low control signal that acts as an input to initialize the CC to a known startup state. The RESET# pin is pulled down internally.

NOTE

The CRG has a built-in RESET# glitch filter to prevent glitches on the RESET# pin from resetting the device (see Section 6.4.1.4, “RESET# Glitch Filter”).

2.4.3.17 INT_CC# — Interrupt Output

INT_CC# is an AMI/MPC and HCS12 interfaces interrupt request output signal. The CC may request a service routine from the host to run. The interrupt is indicated by the logic level: the interrupt is asserted if the INT_CC# outputs a logic 0 and is deasserted if INT_CC# outputs a logic 1.

The pin can be configured to provide high or reduced output drive. This is an open-drain output.

2.4.3.18 TEST

The TEST pin is pulled down, internally, and must be tied to VSS in all applications.

2.4.3.19 DBG[3:2]/CLK_S[1:0] — Debug Strobe Points, Output Clock Select

DBG[3:2] are debug strobe point output signals. The functions output on these pins are selected by the debug port control register. Refer to Section 3.4.16, “Strobe Signal Support” for more information.

NOTE

CLK_S[1:0] signals are inputs during the internal reset sequence and are latched during the internal reset sequence.

While the CLK_S[1:0] levels are being latched, the output drive control is disabled, and the internal pulldown resistors are connected to the pins.

2.4.3.20 EXTAL/CC_CLK — Crystal Driver, External Clock Pin

This pin can act as a crystal driver pin (EXTAL) or as an external clock input pin (CC_CLK). On reset, the device clock is derived from the input frequency on this pin. Refer to Figure 2-3 for Pierce oscillator connections and Figure 2-4 for external clock connections. See also Chapter 7, “Oscillator (OSCV2)”.

2.4.3.21 XTAL — Crystal Driver Pin

XTAL is a crystal driver pin. Refer to Figure 2-3 for oscillator connections and Figure 2-4 for external clock connections. See also Chapter 7, “Oscillator (OSCV2)”.

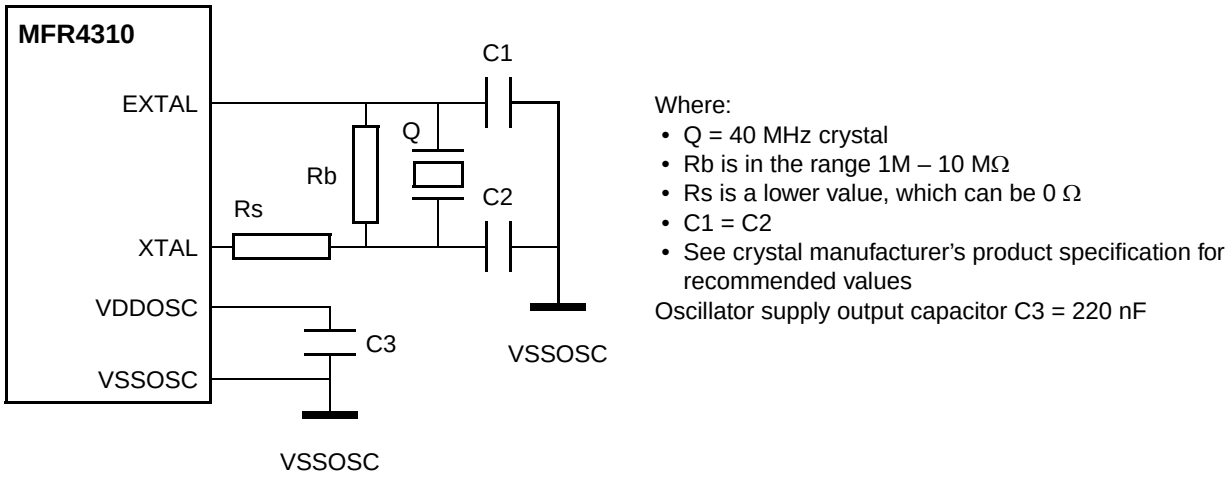


Figure 2-3. Oscillator Connections

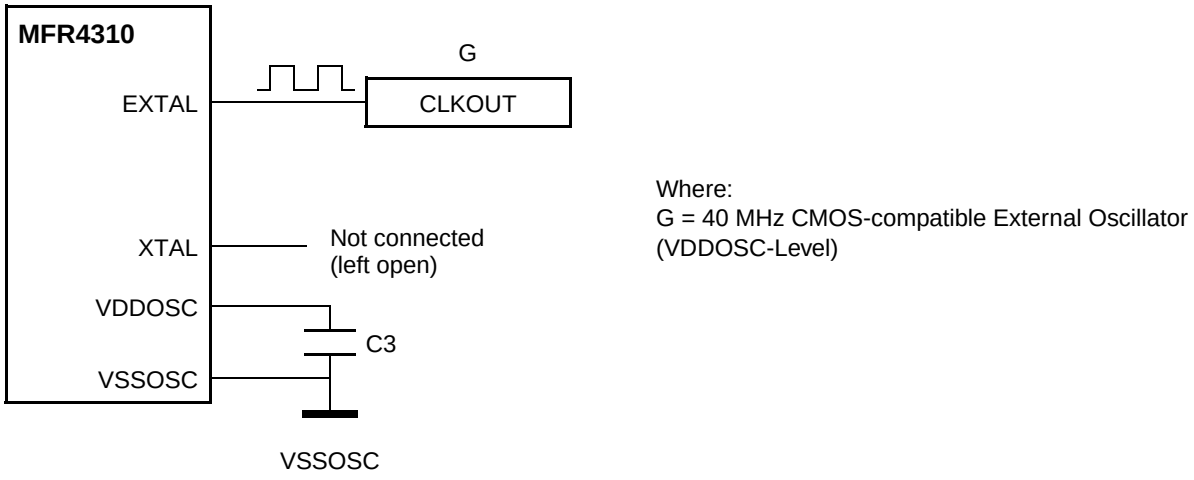


Figure 2-4. External Square Wave Clock Generator Connection

2.4.4 Power Supply Pins

MFR4310 power and ground pins are summarized in Table 2-4 and described below.

NOTE

All VSS pins must be connected together in the application.

Because fast signal transitions place high, short-duration current demands on the power supply, use bypass capacitors with high-frequency characteristics and place them as close to the MFR4310 as possible. Bypass requirements depend on how heavily the MFR4310 pins are loaded.

Table 2-4. MFR4310 Power and Ground Connection Summary

Mnemonic	Pin Number	Nominal Voltage	Description
	64-pin LQFP		
VDD2_5	59	2.5V	Internal power and ground generated by internal regulator
VSS2_5	60	0V	
VDDR	20	3.3V	External power and ground, supply to supply to pin drivers and internal voltage regulator.
VSSR	19	0V	
VDDX[1:4]	8, 37, 54, 35	3.3V	External power and ground, supply to pin drivers.
VSSX[1:4]	9, 38, 53, 31	0V	
VDDA	50	3.3V	Operating voltage and ground for the internal voltage regulator.
VSSA	49	0V	
VDDOSC	26	2.5V	Provides operating voltage and ground for the internal oscillator. This allows the supply voltage to the oscillator to be bypassed independently. Internal power and ground generated by internal regulator.
VSSOSC	23	0V	

2.4.4.1 VDDX, VSSX — Power and Ground Pins for I/O Drivers

External power and ground for I/O drivers.

2.4.4.2 VDDR, VSSR — Power and Ground Pins for I/O Drivers and Internal Voltage Regulator

External power and ground for I/O drivers and input to the internal voltage regulator.

NOTE

The VDDR pin enables the internal 3.3 V to 2.5 V voltage regulator. If this pin is tied to ground, the internal voltage regulator is turned off.

2.4.4.3 VDD2_5, VSS2_5 — Core Power Pins

Power is supplied to the MFR4310 core through VDD2_5 and VSS2_5. This 2.5 V supply is derived from the internal voltage regulator. No static load is allowed on these pins. If VDDR is tied to ground, the internal voltage regulator is turned off.

NOTE

No load is allowed except for bypass capacitors.

2.4.4.4 VDDA, VSSA — Power Supply Pins for VREG

VDDA, VSSA are the power supply and ground input pins for the voltage regulator. They also provide the reference voltages for the internal voltage regulator.

2.4.4.5 VDDOSC, VSSOSC — Power Supply Pins for OSC

VDDOSC, VSSOSC provide operating voltage and ground for the oscillator. This allows the supply voltage to the oscillator to be bypassed independently. This 2.5 V voltage is generated by the internal voltage regulator.

NOTE

No load is allowed except for bypass capacitors.

2.5 Modes of Operation

Refer to Section 3.1.6, “Modes of Operation” for full descriptions of the MFR4310 Disabled and Normal modes of operation.

2.6 External Clock and Host Interface Selection

2.6.1 External 4/10/40 MHz Output Clock

A continuous external 4/10/40 MHz output clock signal is provided by the CC on the CLKOUT pin. See Section 2.4.3.15, “CLKOUT — Clock Output” for details of when this signal is active.

The output frequency of the CLKOUT signal is selected by the CLK_S[1:0] input pins, in accordance with Table 2-5:

Table 2-5. CLKOUT Frequency Selection

Pin		CLKOUT Function
CLK_S0	CLK_S1	
0	0	4 MHz output[1]
1	0	10 MHz output

Table 2-5. CLKOUT Frequency Selection

Pin		CLKOUT Function
CLK_S0	CLK_S1	
0	1	40 MHz output
1	1	Disabled (CLKOUT output is “0“)

1 This is the default clock frequency selection (i.e. if no external pull resistors are connected to CLK_S0 and CLK_S1, the internal pulldown resistors on these pins take effect).

NOTE

As the CLK_S[1:0] signals are multiplexed with DBG[2:3], CLKOUT should be selected using pullup and pulldown resistors

2.6.2 External Host Interface Selection

The MFR4310 can be connected and controlled by two types of interface through the CC EBI. Two pins, IF_SEL0 and IF_SEL1, are used to configure the interface type, in accordance with Table 2-6.

Table 2-6. Interface Selection

Pin		Interface	CHI and Host Interface Clock	CRSR.ECS
IF_SEL0	IF_SEL1			
0	0	MPC Interface	CHICLK_CC	1
0	1	HCS12 Synchronous Interface	CLK_CC	0
1	0	Asynchronous Memory Interface[1]	CLK_CC	0
1	1	Asynchronous Memory Interface	CHICLK_CC	1

1 This is the default interface (i.e. if no external pull resistors are connected to IF_SEL0 and IF_SEL1, the internal pullup on IF_SEL0 and the internal pulldown on IF_SEL1 take effect).

The CC latches the values of the IF_SEL0 and IF_SEL1 signals, when it leaves an internal or external reset state, and analyzes them to configure the interface for the type of external host. The CC does not analyze them after it has left the reset state. For more information on the internal and external reset states, see Chapter 6, “Clocks and Reset Generator (CRG)”.

NOTE

The internal pull devices on IF_SEL1 and IF_SEL0 are enabled only during reset; they are disabled after the reset operation is complete.

NOTE

The following steps must be taken to select a correct external host interface mode.

1. Set IF_SEL0, IF_SEL1 for MPC mode, HCS12 synchronous mode or AMI mode.

2. Assert the external hard reset signal of the CC again.

2.6.3 Recommended Pullup/pulldown Resistor Values

As the IF_SEL[1:0] signals share pins with Physical Layer Interface signals, pullup and pulldown resistors should be used for the selection. The recommended pullup/pulldown resistor values for the IF_SEL[1:0] inputs are given in Table 2-7:

Table 2-7. Recommended Pullup and Pulldown Resistor Values for IF_SEL[1:0] Inputs

IO, Regulator and analog supply level (V_{DD5})	Pullup resistor[1]	Pulldown resistor[1]	Units
3.3V	16	47	kΩ
5V	10	47	kΩ

[1] The listed values are calculated for the MFR4310-Physical Layer connection where no internal pullup/pulldown resistors are assumed in the Electrical PHY at the TXD_BG1 and TXD_BG2 interface lines. If an Electrical PHY device has internal pullup/pulldown resistors connected to these signals, then the external pullup/pulldown resistor values must be recalculated to ensure that V_{IL} requirements for pulldown resistors or V_{IH} requirements for pullup resistors for the chosen V_{DD5} are met. See Section A.1.9, "I/O Characteristics" for more details on V_{IL}, V_{IH} and V_{DD5}.

2.7 External Host Interface

The MFR4310 can be connected through three types of bus interface (see Section 2.6.2, "External Host Interface Selection" for information on how to select the host interface). The three types of microprocessor interface are described below.

2.7.1 Asynchronous Memory Interface

Figure 2-5 shows how to connect the CC to a microcontroller using the AMI interface.

- Data exchange in AMI Mode is controlled by the CE#, WE# and OE# signals.
- The AMI interface is implemented as an asynchronous memory slave module, thus enabling fast interfacing between the CC and a variety of microcontrollers.
- The AMI interface decodes its internal register addresses with the help of the chip select signal CE# and the address lines A[12:1].
- The AMI interface accepts only aligned 16-bit read and 8-bit or 16-bit write transactions. The AMI interface does not support 8-bit read accesses.
  - The byte selects BSEL[1:0]#, the chip enable CE#, the output enable OE#, and the write enable WE# are used to determine the type of access as shown in Table 2-8.

Table 2-8. AMI Access Types

CE#	WE#	OE#	BSEL1#	BSEL0#	Type of Access
0	0	0	X	X	Illegal
0	0	1	0	0	16-bit write to word address[1]
0	0	1	0	1	8-bit write to even byte address[2]

Table 2-8. AMI Access Types

CE#	WE#	OE#	BSEL1#	BSEL0#	Type of Access
0	0	1	1	0	8-bit write to odd byte address[3]
0	0	1	1	1	Illegal
0	1	1	X	X	no access
0	1	0	X	X	16-bit read from word address[4]
1	X	X	X	X	no access

1 Write data from D[15:8] to even byte address and from D[7:0] to odd byte address.
2 Write data from D[15:8].
3 Write data from D[7:0].
4 Read data from even byte address at D[15:8] and from odd byte address at D[7:0].

- WE# indicates the direction of data transfer for a transaction.
- OE# enables the AMI data output to a microcontroller during read transactions.
- INT_CC# is an interrupt line that can be used for requesting, by means of the internal interrupt controller, a service routine from a host controller.
- The AMI interface does not support burst transactions.

NOTE

For the AMI, D0 is the LSB of the 16-bit data bus.

NOTE

If the AMI mode without the CHICLK_CC signal is selected (i.e. IF_SEL[1:0] = 0b01), CHICLK_CC must be driven to logic 0 or logic 1 (it must not be left floating).

2.7.1.1 Asynchronous Memory Interface with S12X Family

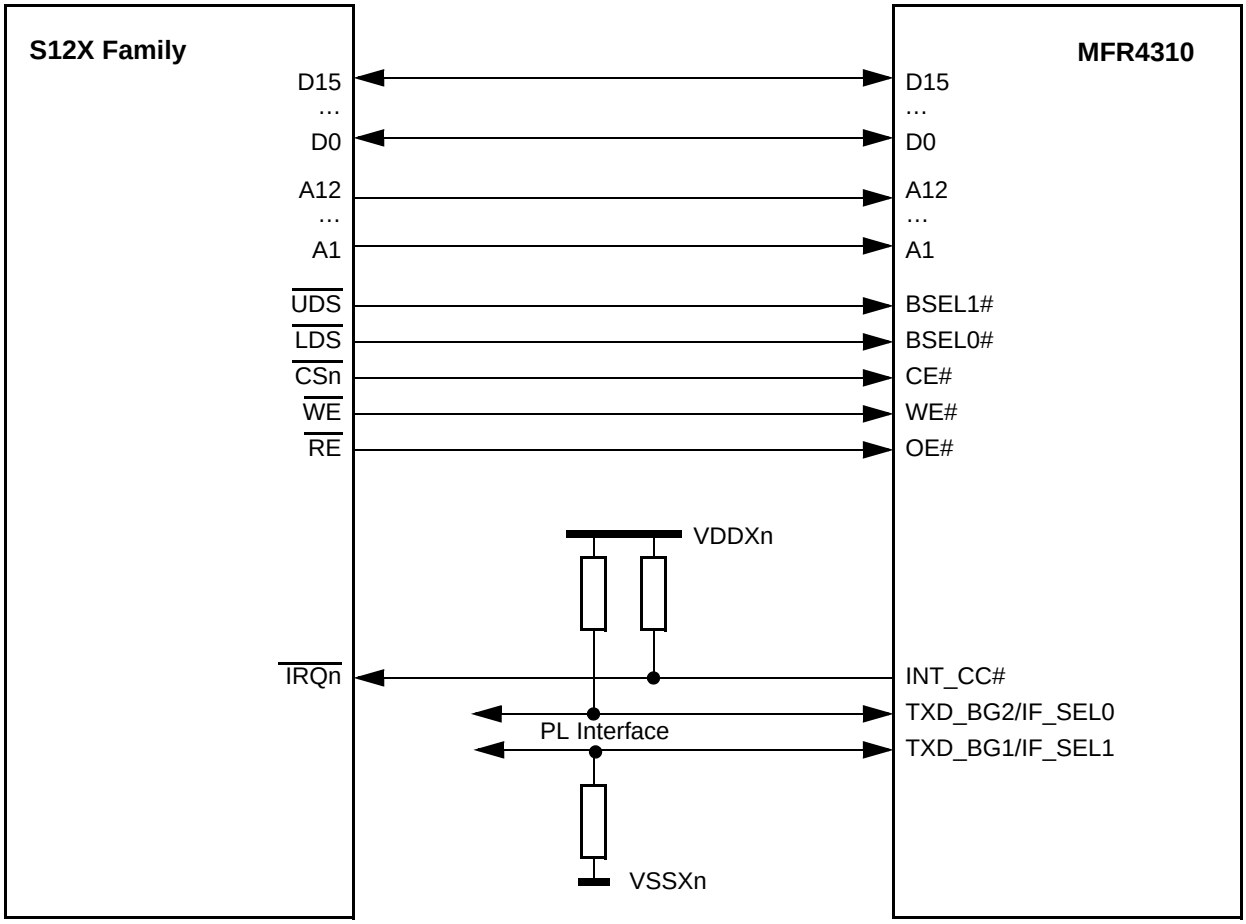


Figure 2-5. AMI Interface with S12X Family

2.7.1.2 Asynchronous Memory Interface with DSP 56F83 (Hawk) Family

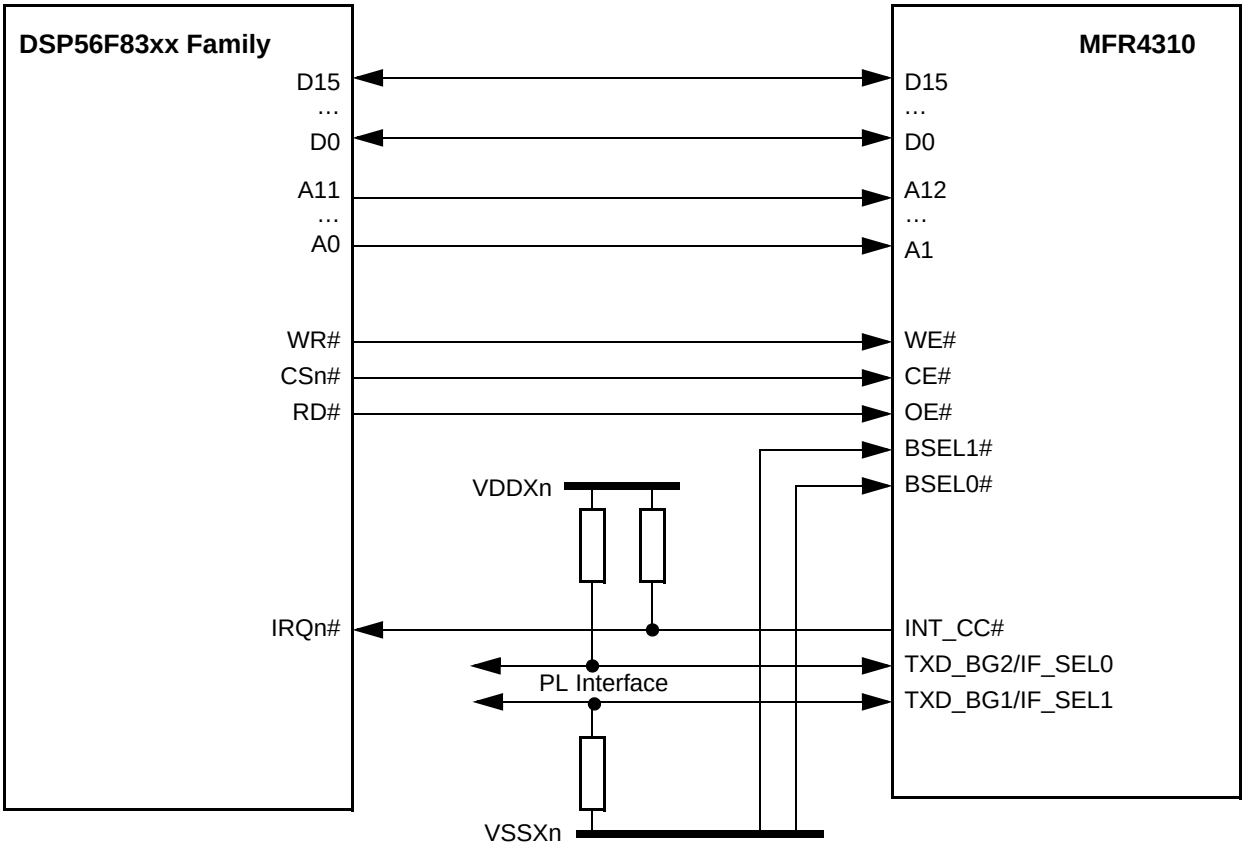


Figure 2-6. AMI Interface with DSP 56F83 (Hawk) Family

2.7.1.3 Asynchronous Memory Interface Timing

See Section A.4, “Asynchronous Memory Interface Timing” for timing characteristics of the AMI interface.

2.7.2 MPC Interface

Figure 2-7 shows how to connect the CC to a microcontroller using the MPC interface. In this case, the host bus pins have the meanings shown in Table 2-9.

- Data exchange in MPC mode is controlled by the CE#, BSEL[1:0]#, and OE# inputs.
- The MPC interface is implemented as an asynchronous memory slave module, thus enabling the fast interfacing with a variety of microcontrollers.
- The MPC interface decodes its internal register addresses with help of the chip select signal CE# and the address lines A[12:1].
- The MPC interface accepts only aligned 16-bit read and 8- or 16-bit write transactions. The MPC interface does not support 8-bit read accesses.

— The chip enable CE#, the output enable OE#, and the write enables BSEL[1:0]# are used to determine the type of access as shown in Table 2-8.

Table 2-9. MPC Interface Access Types

CE#	OE#	BSEL1#	BSEL0#	Type of Access
0	0	X	0	illegal
0	0	0	X	illegal
0	0	1	1	16-bit read from word address[1]
0	1	0	0	16-bit write to word address[2]
0	1	0	1	8-bit write to even byte address[3]
0	1	1	0	8-bit write to odd byte address[4]
0	1	1	1	no access
1	X	X	X	no access

1 Read data from even byte address at D[15:8] and from odd byte address at D[7:0].
2 Write data from D[15:8] to even byte address and from D[7:0] to odd byte address.
3 Write data from D[15:8].
4 Write data from D[7:0].

- BSEL[1:0]# inputs indicate the direction of the data transfer for a transaction.
- OE# input enables the MPC data output during read transactions.

NOTE

D0 is the LSB of the 16-bit data bus.

2.7.2.1 MPC Interface with MPC5xx and MPC55xx Families

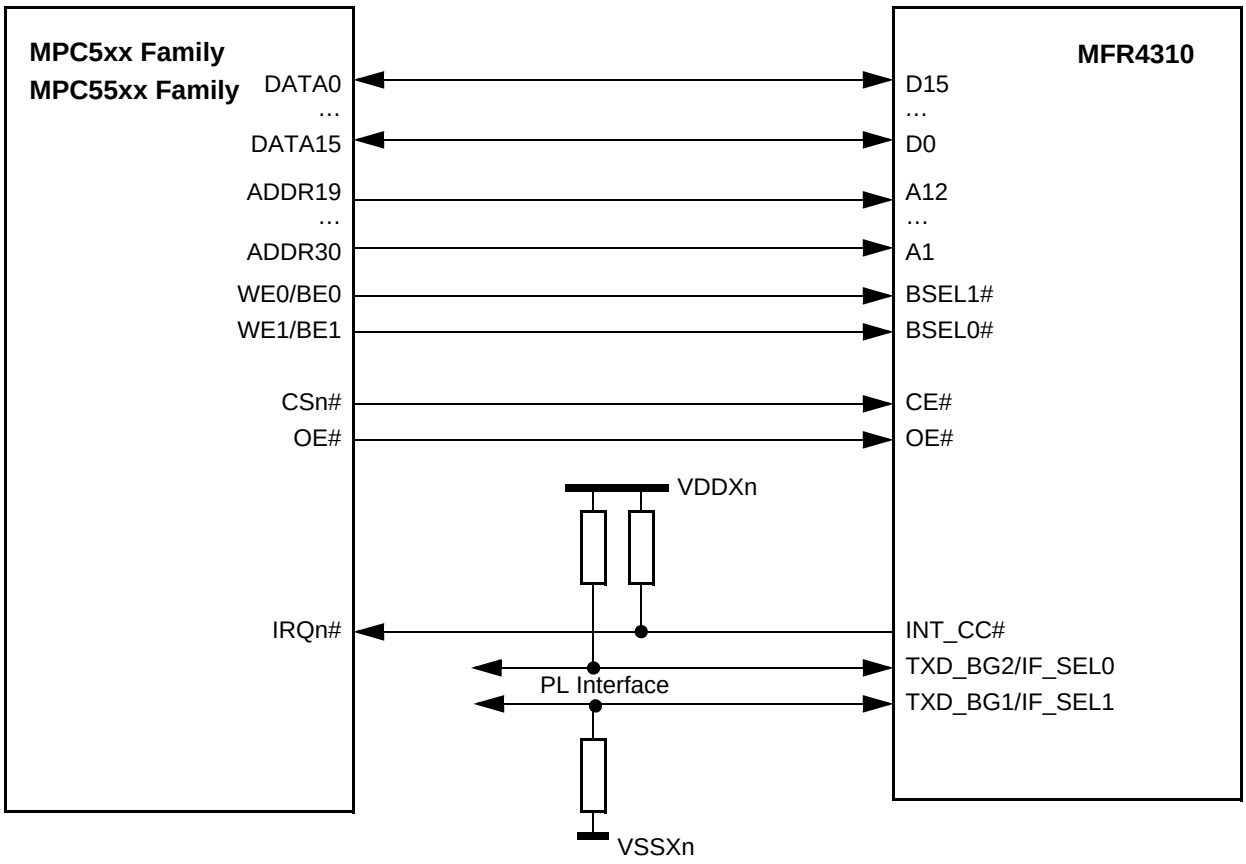


Figure 2-7. MPC EBI Interface with MPC5xx and MPC55xx Families

2.7.2.2 MPC Interface Timing

See Section A.5, “MPC Interface Timing” for timing characteristics of the MPC interface.

2.7.3 HCS12 Interface

Chip selection for the HCS12 interface is generated internally using the following signals (see Figure 2-8):

- The input values of the expanded address signals XADDR[14:19] are compared with logical 0’s (the HCS12 External Bus Interface (EBI) is in the Paged or Unpaged mode).
- The three most significant bits of the demultiplexed address bus, PA[5:7], are compared with the pattern set up externally on the address chip select pins ACS[0:2]; PA5 is compared with ACS0, PA6 with ACS1, PA7 with ACS2.

NOTE

The address decoding phase of a read/write operation is passed if all the comparisons described above are passed.

Figure 2-9 shows how to connect the CC to an HCS12 MCU with EBI paged mode support.

Figure 2-10 shows how to connect he CC to an HCS12 MCU with EBI unpaged mode support.

- The HCS12 interface supports the paged and the unpaged modes of the HCS12 External Bus Interface connected to it.
- The HCS12 interface is implemented as an synchronous HCS12 External Bus slave module, thus enabling the fast data exchange between them.
- The HCS12 interface decodes the addresses of read/write transactions to its internal registers, and generates its internal chip select signal, CS, using the address/data lines PA[0:7], PB[0:7], ACS[0:2], and XADDR[14:19]:
  - The address and data lines PA[0:7], PB[0:7] are multiplexed. They are denoted ADR[0:15] when referring to the address, and DATA[0:15] when referring to the data. The FlexRay CC is selected only when the address ADR[13:15] matches ACS[0:2] (ADR13 matches ACS0, ADR12 matches ACS1, etc.) and the address XADDR[14:19] matches 0.
- The HCS12 interface accepts only aligned 16-bit read and 8-bit or 16-bit write transactions. The HCS12 interface does not support 8-bit read accesses.
  - The internal chip select, CS, the low byte strobe, LSTRB, the least significant bit of the address, ADR0, and the read/write select, RW_CC#, are used to determine the type of access, as shown in Table 2-10.

Table 2-10. HCS12 Access Types

CS	RW_CC#	LSTRB	ADR0	Type of Access
0	X	X	X	No access
1	0	0	0	16-bit write to word address[1]
1	0	0	1	8-bit write to an odd address[2]
1	0	1	0	8-bit write to an even address[2]
1	0	1	1	Not supported
1	1	0	0	16-bit read from an even address[3]
1	1	0	1	Not supported
1	1	1	0	Not supported
1	1	1	1	Not supported

1 Write data from PA to even byte address and from PB to odd byte address.

2 Write data from PB.

3 Read data from even byte address at PA and from odd byte address at PB.

- RW_CC# indicates the direction of data transfer for a transaction.
- INT_CC# is an interrupt line that can be used for requesting, by means of the internal interrupt controller, a service routine from the HCS12 device.

NOTE

AMI-only inputs A[9:7], BSEL[1:0]#/DBG[0:1] (if the debug strobes are disabled), and CHICLK_CC are not used when the HCS12 interface is selected and must be driven to logic 0 or logic 1 (i.e. they must not be left floating).

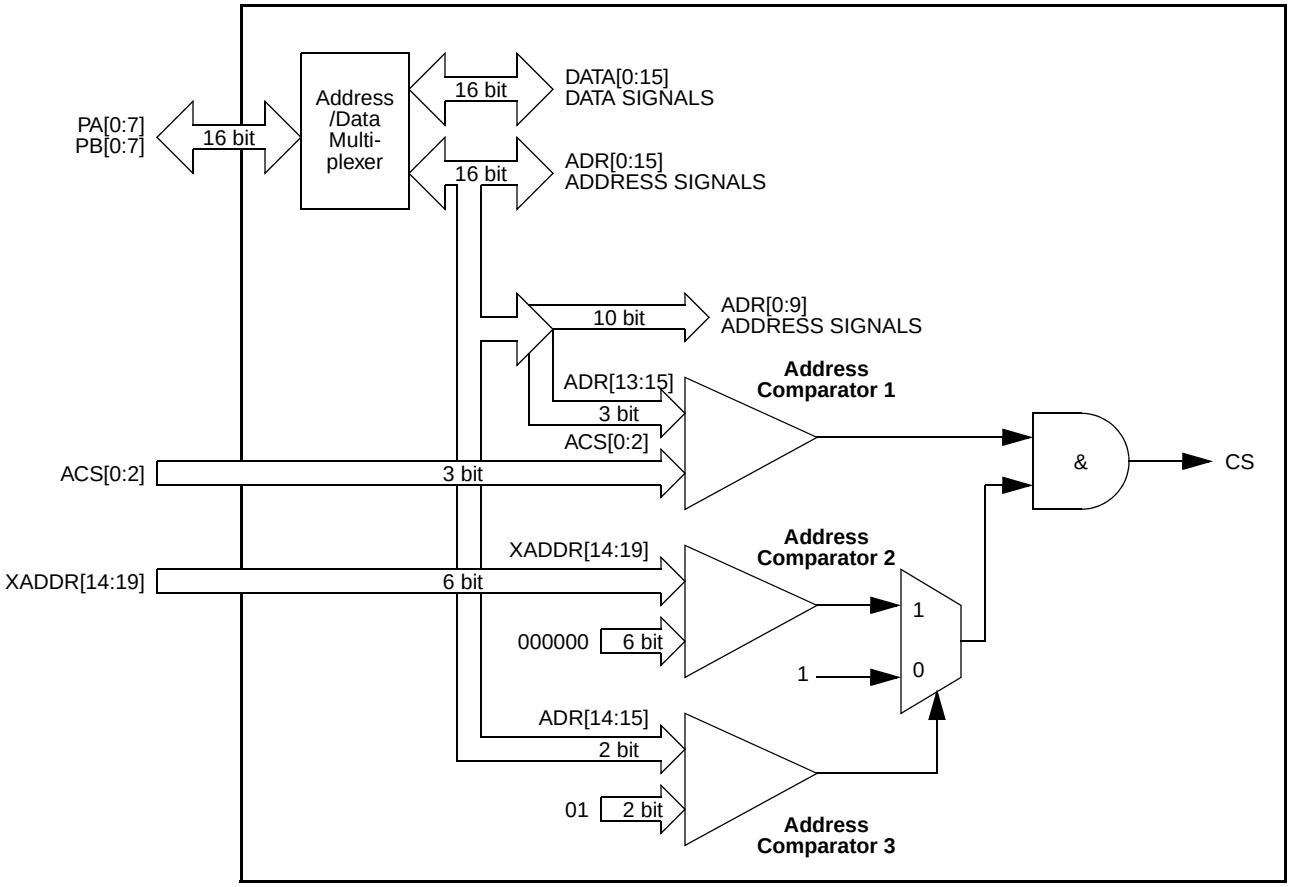


Figure 2-8. HCS12 Interface Address Decoding and Internal Chip Select Generation

2.7.3.1 HCS12 interface with HCS12 Page Mode Support

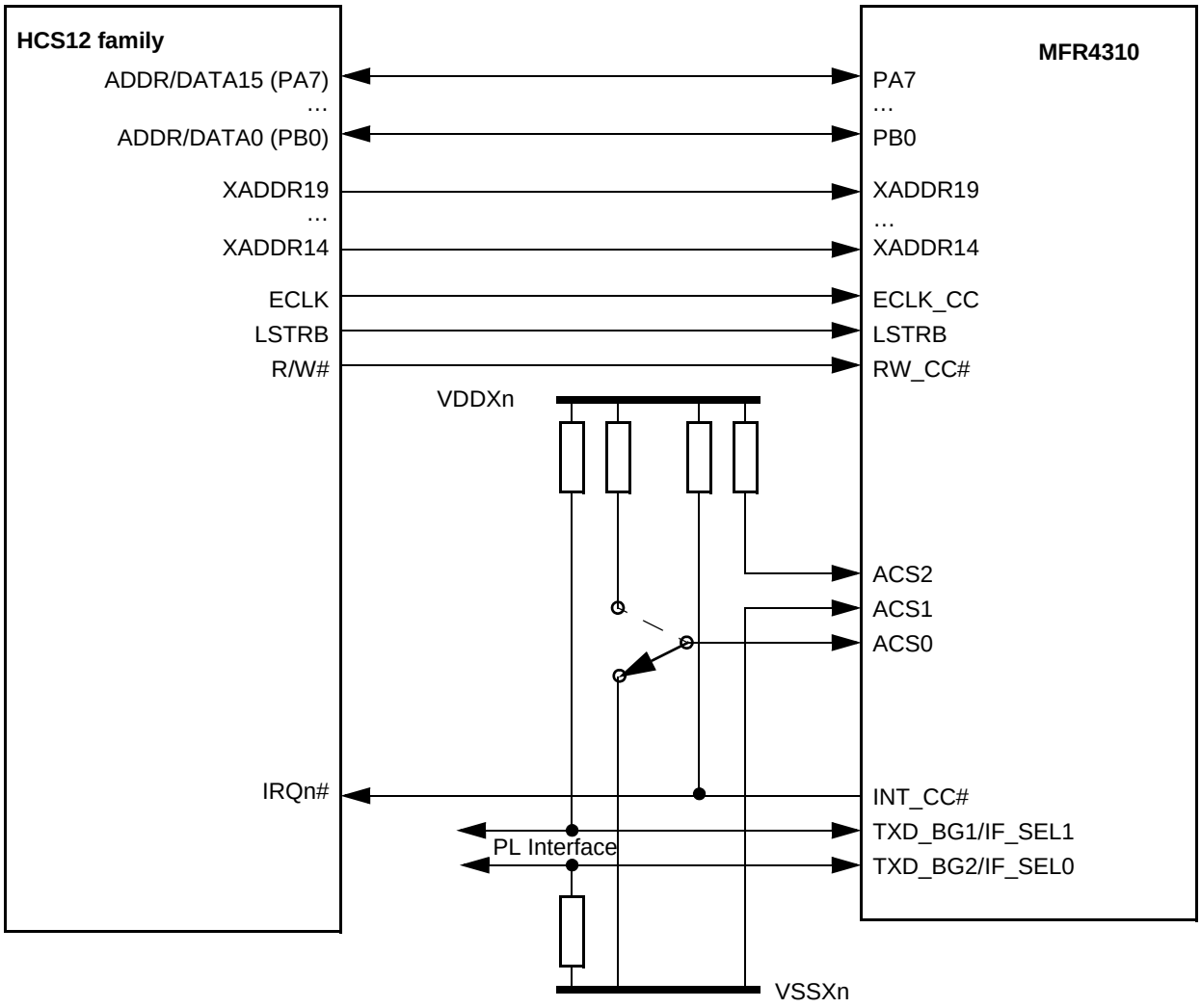


Figure 2-9. HCS12 interface with HCS12 Page Mode Support

2.7.3.2 HCS12 interface with HCS12 Unpaged Mode Support

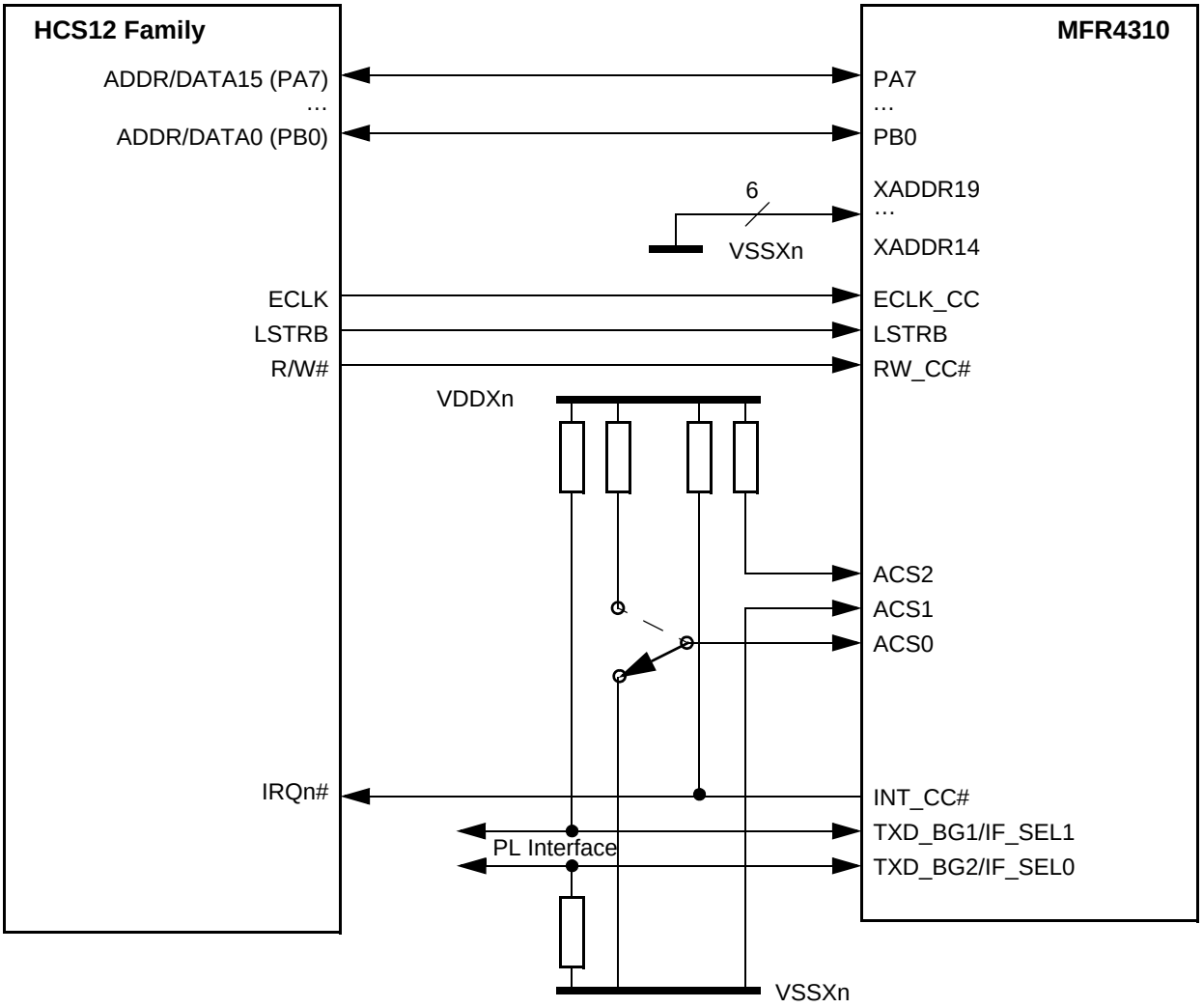


Figure 2-10. HCS12 interface with HCS12 Unpaged Mode Support

2.7.3.3 HCS12 Interface Timing

See Section A.6, “HCS12 Interface Timing” for timing characteristics of the HCS12 interface.

2.8 Resets and Interrupts

2.8.1 Resets

MFR4310 has the following resets:

- External hard reset input signal RESET#.

- Internal power-on and low-voltage resets provided by the internal voltage regulator (refer to Chapter 6, “Clocks and Reset Generator (CRG)” and Chapter 5, “Dual Output Voltage Regulator (VREG3V3V2)” for more information).
- Internal clock monitor failure reset (see Chapter 7, “Oscillator (OSCV2)”).

When a reset occurs, MFR4310 registers and control bits are changed to known startup states. Refer to the respective module chapters for information on the different kinds of resets and for register reset states.

2.8.1.1 I/O Pin States After Reset

Refer to Table 2-3 for the configuration of the MFR4310 pins out of reset.

2.8.2 Interrupt Sources

All possible MFR4310 internal interrupt sources are combined and provided to the host by means of one available interrupt line, INT_CC#. Refer to Section 3.4.19, “Interrupt Support” and Section 6.3.2, “Clock and Reset Status Register (CRSR)” for more information on available interrupt sources. The type of interrupt is level sensitive.

Chapter 3
FlexRay Module (FLEXRAYV4)

3.1 Introduction

3.1.1 Reference

The following documents are referenced.

- *FlexRay Communications System Protocol Specification, Version 2.1 Rev A*
- *FlexRay Communications System Electrical Physical Layer Specification, Version 2.1 Rev A*

3.1.2 Glossary

This section provides a list of terms used in the description of the FlexRay module.

Table 3-1. List of Terms

Term	Definition
BCU	Buffer Control Unit. Handles message buffer access.
CC	Communication Controller
CDC	Clock Domain Crosser
CHI	Controller Host Interface
Cycle length in μT	The actual length of a cycle in μT for the ideal controller (+/- 0 ppm)
EBI	External Bus Interface
FRM	FlexRay Memory. Memory to store message buffer payload, header, and status, and to store synchronization frame related tables.
FSS	Frame Start Sequence
HIF	Host Interface. Provides host access to FlexRay module.
Host	The FlexRay CC host MCU
LUT	Look Up Table. Stores message buffer header index value.
MB	Message Buffer
MBIDX	Message Buffer Index: the position of a header field entry within the header area. If the header area is accessed as an array, this is the same as the array index of the entry.
MBNum	Message Buffer Number: Position of message buffer configuration registers within the register map. For example, Message Buffer Number 5 corresponds to the MBCCS5 register.
MCU	Microcontroller Unit
μT	Microtick. A microtick is one CLK_CC period long, and starts on the rising edge of CLK_CC.
MT	Macrotick
MTS	Media Access Test Symbol

Table 3-1. List of Terms (Continued)

Term	Definition
NIT	Network Idle Time
PE	Protocol Engine
POC	Protocol Operation Control. Each state of the POC is denoted by *POC:state*
Rx	Reception
SEQ	Sequencer Engine
TCU	Time Control Unit
Tx	Transmission

3.1.3 Color Coding

Throughout this chapter types of items are highlighted through the use of an italicized color font.

FlexRay protocol parameters, constants and variables are highlighted with *blue italics*. An example is the parameter *gdActionPointOffset*.

FlexRay protocol states are highlighted in *green italics*. An example is the state *POC:normal active*.

3.1.4 Overview

The FlexRay module is a FlexRay communication controller that implements the *FlexRay Communications System Protocol Specification, Version 2.1 Rev A.*

The FlexRay module has three main components:

- Controller host interface (CHI)
- Protocol engine (PE)
- Clock domain crossing unit (CDC)

A block diagram of the FlexRay module with its surrounding modules is given in Figure 3-1.

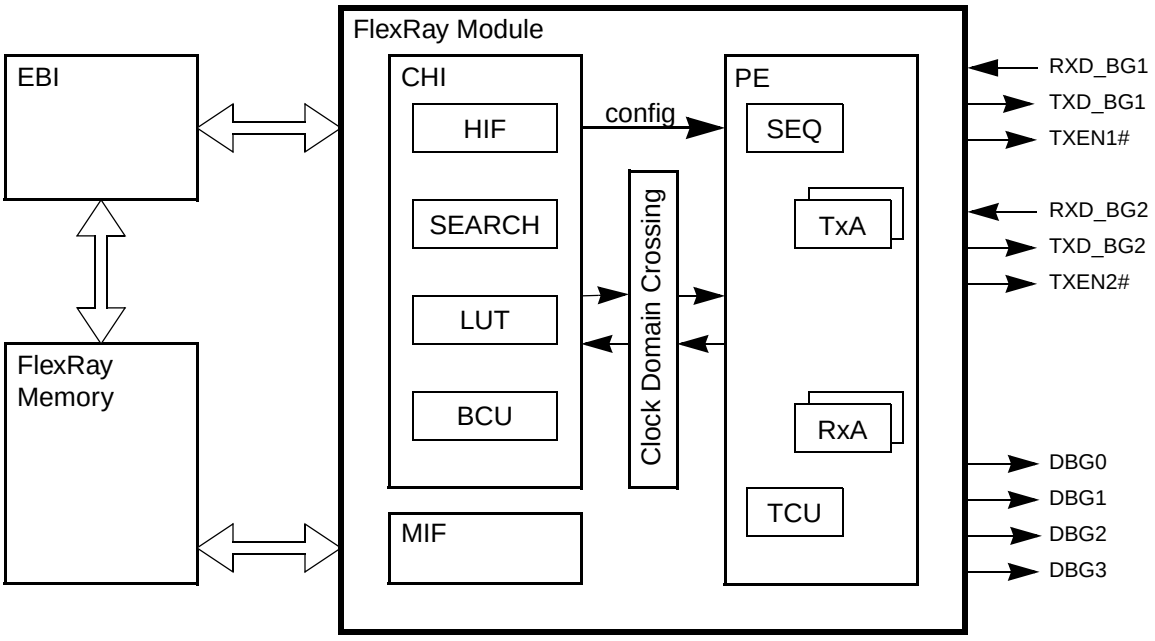


Figure 3-1. FlexRay Module Block Diagram

The protocol engine has two transmitter units TxA and TxB and two receiver units RxA and RxB for sending and receiving frames through the two FlexRay channels. The time control unit (TCU) is responsible for maintaining global clock synchronization to the FlexRay network. The overall activity of the PE is controlled by the sequencer engine (SEQ).

The controller host interface provides host access to the module's configuration, control, and status registers, as well as to the message buffer configuration, control, and status registers. The message buffers themselves, which contain the frame header and payload data received or to be transmitted, and the slot status information, are stored in the FlexRay Memory (FRM).

The clock domain crossing unit implements signal crossing from the CHI clock domain to the PE clock domain and vice versa, to allow for asynchronous PE and CHI clock domains.

The FlexRay module stores the frame header and payload data of frames received or of frames to be transmitted in the FRM. The application accesses the FRM to retrieve and provide the frames to be processed by the FlexRay module. In addition to the frame header and payload data, the FlexRay module stores the synchronization frame related tables in the FRM for application processing.

NOTE

The FlexRay module does not provide a memory protection scheme for the FlexRay Memory.

3.1.5 Features

The FlexRay module provides the following features:

- *FlexRay Communications System Protocol Specification, Version 2.1 Rev A* compliant protocol implementation
- *FlexRay Communications System Electrical Physical Layer Specification, Version 2.1 Rev A* compliant bus driver interface
- single channel support
  - FlexRay Port A can be configured to be connected to physical FlexRay channel A or physical FlexRay channel B.
  - FlexRay bus data rates of 10 Mbit/s, 8 Mbit/s, 5 Mbit/s, and 2.5 Mbit/s supported
- 128 configurable message buffers with
  - individual frame ID filtering
  - individual channel ID filtering
  - individual cycle counter filtering
- message buffer header, status and payload data stored in dedicated FlexRay Memory
  - allows for flexible and efficient message buffer implementation
  - consistent data access ensured by means of buffer locking scheme
  - application can lock multiple buffers at the same time
- size of message buffer payload data section configurable from 0 up to 254 bytes
- two independent message buffer segments with configurable size of payload data section
  - each segment can contain message buffers assigned to the static segment and message buffers assigned to the dynamic segment at the same time
- zero padding for transmit message buffers in static segment
  - applied when the frame payload length exceeds the size of the message buffer data section
- transmit message buffers configurable with state/event semantics
- message buffers can be configured as
  - receive message buffer
  - single buffered transmit message buffer
  - double buffered transmit message buffer (combines two single buffered message buffer)
- individual message buffer reconfiguration supported
  - means provided to safely disable individual message buffers
  - disabled message buffers can be reconfigured
- two independent receive FIFOs
  - one receive FIFO per channel
  - up to 255 entries for each FIFO
  - global frame ID filtering, based on both value/mask filters and range filters
  - global channel ID filtering
  - global message ID filtering for the dynamic segment
- 4 configurable slot error counters
- 4 dedicated slot status indicators
  - used to observe slots without using receive message buffers

- measured value indicatorsfor the clock synchronization
  - internal synchronization frame ID and synchronization frame measurement tables can be copied into the FlexRay Memory
- fractional macroticks aresupported for clock correction
- maskable interrupt sources provided viaindividual and combined interrupt lines
- 1 absolute timer
- 1 timer that can be configured to absolute or relative

3.1.6 Modes of Operation

3.1.6.1 Disabled Mode

This is the default mode the FlexRay module enters during hard reset. The FlexRay module indicates that it is in the Disabled Mode by negating the FlexRay module enable bit MEN in the Module Configuration Register (MCR).

The protocol engine is in its reset state. No communication is performed on the FlexRay bus.

All registers with the write access conditions Any Time and Disabled Mode can be accessed for writing as stated in Section 3.3.2, “Register Descriptions”.

The application can configure the FlexRay module by accessing the FlexRay module configuration bits and fields in the Module Configuration Register (MCR).

The FlexRay module leaves disabled mode when the application sets the FlexRay module enable bit MEN in the Module Configuration Register (MCR) The FlexRay module then deasserts the protocol engine reset and puts the protocol engine into the *POC:default config* state.

NOTE

After the application has enabled the FlexRay module it cannot disable the FlexRay module later on.

3.1.6.2 Normal Mode

In this mode the FlexRay module is fully functional.

The FlexRay module indicates that it is in normal mode by asserting the FlexRay module enable bit (MEN) in the Module Configuration Register (MCR).

This mode is entered when the application requests the FlexRay module to leave the disabled mode. If this mode is entered, the protocol engine is in its *POC:default config* state.

Depending on the values of the SCM, CHA, and CHB bits in the Module Configuration Register (MCR), the corresponding FlexRay bus driver ports are enabled and driven.

The application can transition the protocol engine into other protocol states using the Protocol Operation Control Register (POCR). For details regarding protocol states, see *FlexRay Communications System Protocol Specification, Version 2.1 Rev A.*

3.2 External Signal Description

This section lists and describes the FlexRay module signals, connected to external pins. These signals are summarized in Table 3-2 and described in detail in Section 3.2.1, “Detailed Signal Descriptions”.

NOTE

The off chip signals RXD_BG1, TXD_BG1, and TXEN1# are available on each package option. The availability of the other off chip signals depends on the package option.

Table 3-2. External Signal Properties

Name	Direction	Active	Reset	Function
EXTAL/CLK_CC	Input	—	—	External Protocol Engine Clock
RXD_BG1	Input	—	—	Receive Data Channel A
TXD_BG1	Output	—	1	Transmit Data Channel A
TXEN1#	Output	Low	1	Transmit Enable Channel A
RXD_BG2	Input	—	—	Receive Data Channel B
TXD_BG2	Output	—	1	Transmit Data Channel B
TXEN2#	Output	Low	1	Transmit Enable Channel B
DBG0	Output	—	0	Debug Strobe Signal 0
DBG1	Output	—	0	Debug Strobe Signal 1
DBG2	Output	—	0	Debug Strobe Signal 2
DBG3	Output	—	0	Debug Strobe Signal 3

3.2.1 Detailed Signal Descriptions

This section provides a detailed description of the FlexRay module signals, connected to external pins.

3.2.1.1 EXTAL/CLK_CC — External Protocol Engine Clock

The EXTAL/CLK_CC signal carries the 40 MHz clock source signal for the Protocol Engine clock.

3.2.1.2 RXD_BG1 — Receive Data Channel A

The RXD_BG1 signal carries the receive data for channel A from the corresponding FlexRay bus driver.

3.2.1.3 TXD_BG1 — Transmit Data Channel A

The TXD_BG1 signal carries the transmit data for channel A to the corresponding FlexRay bus driver.

3.2.1.4 TXEN1# — Transmit Enable Channel A

The TXEN1# signal indicates to the FlexRay bus driver that the FlexRay module is attempting to transmit data on channel A.

3.2.1.5 RXD_BG2 — Receive Data Channel B

The RXD_BG2 signal carries the receive data for channel B from the corresponding FlexRay bus driver.

3.2.1.6 TXD_BG2 — Transmit Data Channel B

The TXD_BG2 signal carries the transmit data for channel B to the corresponding FlexRay bus driver

3.2.1.7 TXEN2# — Transmit Enable Channel B

The TXEN2# signal indicates to the FlexRay bus driver that the FlexRay module is attempting to transmit data on channel B.

3.2.1.8 DBG3, DBG2, DBG1, DBG0 — Strobe Signals

These signals provide the selected debug strobe signals. For details on the debug strobe signal selection refer to Section 3.4.16, “Strobe Signal Support”.

3.3 Memory Map and Register Description

The FlexRay module occupies 1280 bytes of address space starting at address 0x0000.

3.3.1 Memory Map

The complete memory map of the FlexRay module is shown in Table 3-3.

Table 3-3. FlexRay Memory Map

Address	Register	Access
Module Configuration and Control		
0x0000	Module Version Register (MVR)	R
0x0002	Module Configuration Register (MCR)	R/W
0x0004	Reserved	R
0x0006	Reserved	R
0x0008	Strobe Signal Control Register (STBSCR)	R/W
0x000A	Reserved	R/W
0x000C	Message Buffer Data Size Register (MBDSR)	R/W
0x000E	Message Buffer Segment Size and Utilization Register (MBSSUTR)	R/W
Test Registers		
0x0010	Reserved	R
0x0012	Reserved	R
Interrupt and Error Handling		
0x0014	Protocol Operation Control Register (POCR)	R/W
0x0016	Global Interrupt Flag and Enable Register (GIFER)	R/W
0x0018	Protocol Interrupt Flag Register 0 (PIFR0)	R/W
0x001A	Protocol Interrupt Flag Register 1 (PIFR1)	R/W
0x001C	Protocol Interrupt Enable Register 0 (PIER0)	R/W

Table 3-3. FlexRay Memory Map (Continued)

Address	Register	Access
0x001E	Protocol Interrupt Enable Register 1 (PIER1)	R/W
0x0020	CHI Error Flag Register (CHIERFR)	R/W
0x0022	Message Buffer Interrupt Vector Register (MBIVEC)	R
0x0024	Channel A Status Error Counter Register (CASERCR)	R
0x0026	Channel B Status Error Counter Register (CBSERCR)	R
Protocol Status		
0x0028	Protocol Status Register 0 (PSR0)	R
0x002A	Protocol Status Register 1 (PSR1)	R
0x002C	Protocol Status Register 2 (PSR2)	R
0x002E	Protocol Status Register 3 (PSR3)	R/W
0x0030	Macrotick Counter Register (MTCTR)	R
0x0032	Cycle Counter Register (CYCTR)	R
0x0034	Slot Counter Channel A Register (SLTCTAR)	R
0x0036	Slot Counter Channel B Register (SLTCTBR)	R
0x0038	Rate Correction Value Register (RTCORVR)	R
0x003A	Offset Correction Value Register (OFCORVR)	R
0x003C	Combined Interrupt Flag Register (CIFRR)	R
0x003E	Reserved	R
Sync Frame Counter and Tables		
0x0040	Sync Frame Counter Register (SFCNTR)	R
0x0042	Sync Frame Table Offset Register (SFTOR)	R/W
0x0044	Sync Frame Table Configuration, Control, Status Register (SFTCCSR)	R/W
Sync Frame Filter		
0x0046	Sync Frame ID Rejection Filter Register (SFIDRFR)	R/W
0x0048	Sync Frame ID Acceptance Filter Value Register (SFIDAFVR)	R/W
0x004A	Sync Frame ID Acceptance Filter Mask Register (SFIDAFMR)	R/W
Network Management Vector		
0x004C	Network Management Vector Register 0 (NMVR0)	R
0x004E	Network Management Vector Register 1 (NMVR1)	R
0x0050	Network Management Vector Register 2 (NMVR2)	R
0x0052	Network Management Vector Register 3 (NMVR3)	R
0x0054	Network Management Vector Register 4 (NMVR4)	R
0x0056	Network Management Vector Register 5 (NMVR5)	R
0x0058	Network Management Vector Length Register (NMVLR)	R/W
Timer Configuration		
0x005A	Timer Configuration and Control Register (TICCR)	R/W
0x005C	Timer 1 Cycle Set Register (TI1CYSR)	R/W
0x005E	Timer 1 Macrotick Offset Register (TI1MTOR)	R/W
0x0060	Timer 2 Configuration Register 0 (TI2CR0)	R/W
0x0062	Timer 2 Configuration Register 1 (TI2CR1)	R/W
Slot Status Configuration		

Table 3-3. FlexRay Memory Map (Continued)

Address	Register	Access
0x0064	Slot Status Selection Register (SSSR)	R/W
0x0066	Slot Status Counter Condition Register (SSCCR)	R/W
Slot Status		
0x0068	Slot Status Register 0 (SSR0)	R
0x006A	Slot Status Register 1 (SSR1)	R
0x006C	Slot Status Register 2 (SSR2)	R
0x006E	Slot Status Register 3 (SSR3)	R
0x0070	Slot Status Register 4 (SSR4)	R
0x0072	Slot Status Register 5 (SSR5)	R
0x0074	Slot Status Register 6 (SSR6)	R
0x0076	Slot Status Register 7 (SSR7)	R
0x0078	Slot Status Counter Register 0 (SSCR0)	R
0x007A	Slot Status Counter Register 1 (SSCR1)	R
0x007C	Slot Status Counter Register 2 (SSCR2)	R
0x007E	Slot Status Counter Register 3 (SSCR3)	R
MTS Generation		
0x0080	MTS A Configuration Register (MTSACFR)	R/W
0x0082	MTS B Configuration Register (MTSBCFR)	R/W
Shadow Buffer Configuration		
0x0084	Receive Shadow Buffer Index Register (RSBIR)	R/W
Receive FIFO — Configuration		
0x0086	Receive FIFO Selection Register (RFSR)	R/W
0x0088	Receive FIFO Start Index Register (RFSIR)	R/W
0x008A	Receive FIFO Depth and Size Register (RFDSR)	R/W
Receive FIFO - Status		
0x008C	Receive FIFO A Read Index Register (RFARIR)	R
0x008E	Receive FIFO B Read Index Register (RFBRIR)	R
Receive FIFO - Filter		
0x0090	Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR)	R/W
0x0092	Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR)	R/W
0x0094	Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR)	R/W
0x0096	Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR)	R/W
0x0098	Receive FIFO Range Filter Configuration Register (RFRFCFR)	R/W
0x009A	Receive FIFO Range Filter Control Register (RFRFCTR)	R/W
Dynamic Segment Status		
0x009C	Last Dynamic Transmit Slot Channel A Register (LDTXSLAR)	R
0x009E	Last Dynamic Transmit Slot Channel B Register (LDTXSLBR)	R
Protocol Configuration		

Table 3-3. FlexRay Memory Map (Continued)

Address	Register	Access
0x00A0 ... 0x00DC	Protocol Configuration Register 0 (PCR0) ... Protocol Configuration Register 30 (PCR30)	R/W – R/W
0x00DE ... 0x00FE	Reserved	R
Message Buffers Configuration, Control, Status		
0x0100	Message Buffer Configuration, Control, Status Register 0 (MBCCSR0)	R/W
0x0102	Message Buffer Cycle Counter Filter Register 0 (MBCCFR0)	R/W
0x0104	Message Buffer Frame ID Register 0 (MBFIDR0)	R/W
0x0106	Message Buffer Index Register 0 (MBIDXR0)	R/W
...	...	...
0x04F8	Message Buffer Configuration, Control, Status Register 127 (MBCCSR127)	R/W
0x04FA	Message Buffer Cycle Counter Filter Register 127 (MBCCFR127)	R/W
0x04FC	Message Buffer Frame ID Register 127 (MBFIDR127)	R/W
0x04FE	Message Buffer Index Register 127 (MBIDXR127)	R/W

3.3.2 Register Descriptions

This section provides detailed descriptions of all registers in ascending address order, presented as 16-bit wide entities.

Table 3-4 provides a key for the register figures and register tables.

Table 3-4. Register Access Conventions

Convention	Description
	The shaded field indicates that the bit or field is not writeable.
R*	The R* item indicates a reserved bit or field. The FlexRay module does not change its value. The application must not write any value different from the reset value to this bit or field.
Reset Value	
0	Resets to zero.
1	Resets to one.
–	Not defined after reset and not affected by reset.

3.3.2.1 Register Reset

All registers except the Message Buffer Cycle Counter Filter Registers (MBCCFRn), Message Buffer Frame ID Registers (MBFIDRn), and Message Buffer Index Registers (MBIDXRn) are reset to their reset value on system reset. The registers mentioned above are located in physical memory blocks and, thus, they are not affected by reset. For some register fields, additional reset conditions exist. These additional reset conditions are mentioned in the detailed description of the register. The additional reset conditions are explained in Table 3-5.

Table 3-5. Additional Register Reset Conditions

Condition	Description
Protocol RUN Command	The register field is reset when the application writes to RUN command “0101” to the POCCMD field in the Protocol Operation Control Register (POCR).
Message Buffer Disable	The register field is reset when the application has disabled the message buffer. This happens when the application writes 1 to the message buffer disable trigger bit MBCCSRn.EDT while the message buffer is enabled (MBCCSn.EDS = 1) and the FlexRay module grants the disable to the application by clearing the MBCCSRn.EDS bit.

3.3.2.2 Register Write Access

This section describes the write access restriction terms that apply to all registers.

3.3.2.2.1 Register Write Access Restriction

For each register bit and register field, the write access conditions are specified in the detailed register description. A description of the write access conditions is given in Table 3-6. If, for a specific register bit or field, none of the given write access conditions is fulfilled, any write attempt to this register bit or field is ignored without any notification. The values of the bits or fields are not changed. The condition term [A or B] indicates that the register or field can be written to if at least one of the conditions is fulfilled.

Table 3-6. Register Write Access Restrictions

Condition	Indication	Description
Any Time	-	No write access restriction.
Disabled Mode	MCR.MEN = 0	Write access only when the FlexRay module is in Disabled Mode.
Normal Mode	MCR.MEN = 1	Write access only when the FlexRay module is in Normal Mode.
POC:config	PSR0.PROTSTATE = *POC:config*	Write access only when the Protocol is in the *POC:config* state.
MB_DIS	MBCCSRn.EDS = 0	Write access only when the related Message Buffer is disabled.
MB_LCK	MBCCSRn.LCKS = 1	Write access only when the related Message Buffer is locked.

3.3.2.2.2 Register Write Access Requirements

For some of the registers, a 16-bit wide write access is required to ensure correct operation. This write access requirement is stated in the detailed register description for each register affected

3.3.2.2.3 Internal Register Access

The following memory mapped registers are used to access multiple internal registers.

- Strobe Signal Control Register (STBSCR)
- Slot Status Selection Register (SSSR)
- Slot Status Counter Condition Register (SSCCR)
- Receive Shadow Buffer Index Register (RSBIR)

Each of these memory mapped registers provides a SEL field and a WMD bit. The SEL field is used to select the internal register. The WMD bit controls the write mode. If the WMD bit is set to 0 during the write access, all fields of the internal register are updated. If the WMD bit set to 1, only the SEL field is

changed. All other fields of the internal register remain unchanged. This allows for reading back the values of the selected internal register in a subsequent read access.

3.3.2.3 Module Version Register (MVR)

0x0000

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHIVER								PEVER							
W																
Reset	1	0	0	0	0	1	0	1	0	1	1	0	0	1	1	0

Figure 3-2. Module Version Register (MVR)

This register provides the FlexRay module version number. The module version number is derived from the CHI version number and the PE version number.

Table 3-7. MVR Field Descriptions

Field	Description
15–8 CHIVER	**CHI Version Number** — This field provides the version number of the controller host interface.
7–0 PEVER	**PE Version Number** — This field provides the version number of the protocol engine.

3.3.2.4 Module Configuration Register (MCR)

0x0002

Write: MEN, SCM, CHB, CHA, BITRATE: Disabled Mode
SFFE: Disabled Mode or *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MEN	0	SCM	CHB	CHA	SFFE	0	R*	0	0	0	R*	BITRATE			0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-3. Module Configuration Register (MCR)

This register defines the global configuration of the FlexRay module.

Table 3-8. MCR Field Descriptions

Field	Description
15 MEN	**Module Enable** — This bit indicates whether or not the FlexRay module is in the Disabled Mode. The application requests the FlexRay module to leave the Disabled Mode by writing 1 to this bit. Before leaving the Disabled Mode, the application must configure the SCM, CHB, CHA, TMODE, BITRATE values. For details see Section 3.1.6, “Modes of Operation”. 0 Write: ignored, FlexRay module disable not possible Read: FlexRay module disabled 1 Write: enable FlexRay module Read: FlexRay module enabled **Note:** If the FlexRay module is enabled it can not be disabled.
13 SCM	**Single Channel Device Mode** — This control bit defines the channel device mode of the FlexRay module as described in Section 3.4.10, “Channel Device Modes”. 0 FlexRay module works in dual channel device mode 1 FlexRay module works in single channel device mode
12–11 CHB CHA	**Channel Enable** — protocol related parameter: *pChannels* The semantic of these control bits depends on the channel device mode controlled by the SCM bit and is given Table 3-9.
10 SFFE	**Synchronization Frame Filter Enable** — This bit controls the filtering for received synchronization frames. For details see Section 3.4.15, “Sync Frame Filtering”. 0 Synchronization frame filtering disabled 1 Synchronization frame filtering enabled
8 R*	**Reserved** — This bit is reserved. It is read as 0. Application must not write 1 to this bit.
4 R*	**Reserved** — This bit is reserved. It is read as 0. Application must not write 1 to this bit.
3–1 BITRATE	**FlexRay Bus Bit Rate** — This bit field defines the bit rate of the flexray channels according to Table 3-10.

Table 3-9. FlexRay Channel Selection

SCM	CHB	CHA	Description
Dual Channel Device Modes			
0	0	0	ports RXD_BG1, TXD_BG1, and TXEN1# not driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# not driven by FlexRay module PE channel 0 idle PE channel 1 idle
	0	1	ports RXD_BG1, TXD_BG1, and TXEN1# driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# not driven by FlexRay module PE channel 0 active PE channel 1 idle
	1	0	ports RXD_BG1, TXD_BG1, and TXEN1# not driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# driven by FlexRay module PE channel 0 idle PE channel 1 active
	1	1	ports RXD_BG1, TXD_BG1, and TXEN1# driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# driven by FlexRay module PE channel 0 active PE channel 1 active
Single Channel Device Mode			

Table 3-9. FlexRay Channel Selection (Continued)

SCM	CHB	CHA	Description
1	0	0	ports RXD_BG1, TXD_BG1, and TXEN1# not driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# not driven by FlexRay module PE channel 0 idle PE channel 1 idle
	0	1	ports RXD_BG1, TXD_BG1, and TXEN1# driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# not driven by FlexRay module PE channel 0 active PE channel 1 idle
	1	0	ports RXD_BG1, TXD_BG1, and TXEN1# driven by FlexRay module ports RXD_BG2, TXD_BG2, and TXEN1# not driven by FlexRay module PE channel 0 active, uses cCrcInit[B] (see Figure 3-131) PE channel 1 idle
	1	1	reserved

Table 3-10. FlexRay Channel Bit Rate Selection

MCR [BITRATE]	FlexRay Channel Bit Rate [Mbit/s]
000	10.0
001	5.0
010	2.5
011	8.0
100	reserved
101	reserved
110	reserved
111	reserved

3.3.2.5 Strobe Signal Control Register (STBSCR)

0x0008 — 16-bit write access required — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	SEL							0	0	0	ENB	0	0	STBPSEL	
W	WMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-4. Strobe Signal Control Register (STBSCR)

This register is used to assign the individual protocol timing related strobe signals given in Table 3-12 to the external strobe ports. Each strobe signal can be assigned to at most one strobe port. Each write access to registers overwrites the previously written ENB and STBPSEL values for the signal indicated by SEL. If more than one strobe signal is assigned to one strobe port, the current values of the strobe signals are combined with a binary OR and presented at the strobe port. If no strobe signal is assigned to a strobe port, the strobe port carries logic 0. For more detailed and timing information refer to Section 3.4.16, “Strobe Signal Support”.

NOTE

In single channel device mode, channel B related strobe signals are undefined and should not be assigned to the strobe ports.

Table 3-11. STBSCR Field Descriptions

Field	Description
15 WMD	**Write Mode** — This control bit defines the write mode of this register. 0 Write to all fields in this register on write access. 1 Write to SEL field only on write access.
14–8 SEL	**Strobe Signal Select** — This control field selects one of the strobe signals given in Table 3-12 to be enabled or disabled and assigned to one of the four strobe ports given in Table 3-12.
4 ENB	**Strobe Signal Enable** — The control bit is used to enable and to disable the strobe signal selected by STBSSEL. 0 Strobe signal is disabled and not assigned to any strobe port. 1 Strobe signal is enabled and assigned to the strobe port selected by STBPSEL.
1–0 STBPSEL	**Strobe Port Select** — This field selects the strobe port that the strobe signal selected by the SEL is assigned to. All strobe signals that are enabled and assigned to the same strobe port are combined with a binary OR operation. 00 assign selected signal to DBG0 01 assign selected signal to DBG1 10 assign selected signal to DBG2 11 assign selected signal to DBG3

Table 3-12. Strobe Signal Mapping

SEL		Description	Channel	Type	Offset[1]	Reference
dec	hex					
0	0x00	poc_startup_state[0] (for coding see PSR0[4])	-	value	0	MT start
1	0x01	poc_startup_state[1] (for coding see PSR0[5])				
2	0x02	poc_startup_state[2] (for coding see PSR0[6])				
3	0x03	poc_startup_state[3] (for coding see PSR0[7])				
4	0x04	poc_state[0] (for coding see PSR0[8])				
5	0x05	poc_state[1] (for coding see PSR0[9])				
6	0x06	poc_state[2] (for coding see PSR0[10])				
7	0x07	channel idle indicator	A	level	+5	RXD_BG1
8	0x08		B			RXD_BG2
9	0x09	receive data after glitch filtering	A	value	+4	RXD_BG1
10	0x0A		B			RXD_BG2
11	0x0B	synchronization edge strobe	A	pulse	+4	RXD_BG1
12	0x0C		B			RXD_BG2
13	0x0D	header received	A	pulse	+4	RXD_BG1
14	0x0E		B			RXD_BG2
15	0x0F	wakeup symbol decoded	A	pulse	+5	RXD_BG1
16	0x10		B			RXD_BG2
17	0x11	MTS or CAS symbol decoded	A	pulse	+4	RXD_BG1
18	0x12		B			RXD_BG2
19	0x13	frame decoded	A	pulse	+4	RXD_BG1
20	0x14		B			RXD_BG2
21	0x15	channel idle detected	A	pulse	+4	RXD_BG1
22	0x16		B			RXD_BG2

Table 3-12. Strobe Signal Mapping (Continued)

SEL		Description	Channel	Type	Offset[1]	Reference
dec	hex					
23	0x17	start of communication element detected	A	pulse	+4	RXD_BG1
24	0x18		B			RXD_BG2
25	0x19	potential frame start channel	A	pulse	+4	RXD_BG1
26	0x1A		B			RXD_BG2
27	0x1B	wakeup collision detected	A	pulse	+5	RXD_BG1
28	0x1C		B			RXD_BG2
29	0x1D	content error detected	A	level	+4	RXD_BG1
30	0x1E		B			RXD_BG2
31	0x1F	syntax error detected	A	pulse	+4	RXD_BG1
32	0x20		B			RXD_BG2
33	0x21	start transmission of wakeup pattern	A	pulse	-1	TXD_BG1
34	0x22		B			TXD_BG2
35	0x23	start transmission of MTS or CAS symbol	A	pulse	-1	TXD_BG1
36	0x24		B			TXD_BG2
37	0x25	start of transmission	A	pulse	-1	TXD_BG1
38	0x26		B			TXD_BG2
39	0x27	end of transmission	A	pulse	-1	TXD_BG1
40	0x28		B			TXD_BG2
41	0x29	static segment indicator	-	level	0	MT start
42	0x2A	dynamic segment indicator	-	level	0	MT start
43	0x2B	symbol window indicator	-	level	0	MT start
44	0x2C	NIT indicator	-	level	0	MT start
45	0x2D	action point	-	pulse	-1	TXD_BG1
46	0x2E	sync calculation complete[2]	-	pulse	-	-
47	0x2F	start of offset correction	-	pulse	-2	MT start
48	0x30	cycle count[0]	-	value	-2	MT start
49	0x31	cycle count[1]				
50	0x32	cycle count[2]				
51	0x33	cycle count[3]				
52	0x34	cycle count[4]				
53	0x35	cycle count[5]				

Table 3-12. Strobe Signal Mapping (Continued)

SEL		Description	Channel	Type	Offset[1]	Reference
dec	hex					
54	0x36	slot count[0]	A	value	0	MT start
55	0x37	slot count[1]				
56	0x38	slot count[2]				
57	0x39	slot count[3]				
58	0x3A	slot count[4]				
59	0x3B	slot count[5]				
60	0x3C	slot count[6]				
61	0x3D	slot count[7]				
62	0x3E	slot count[8]				
63	0x3F	slot count[9]				
64	0x40	slot count[10]				
65	0x41	slot count[0]	B	value	0	MT start
66	0x42	slot count[1]				
67	0x43	slot count[2]				
68	0x44	slot count[3]				
69	0x45	slot count[4]				
70	0x46	slot count[5]				
71	0x47	slot count[6]				
72	0x48	slot count[7]				
73	0x49	slot count[8]				
74	0x4A	slot count[9]				
75	0x4B	slot count[10]				
76	0x4C	cycle start	-	pulse	0	MT start
77	0x4D	slot start	A	pulse	0	MT start
78	0x4E		B			
79	0x4F	minislot start	-	pulse	0	MT start
80	0x50	arm	-	value	+1	MT start
81	0x51	mt	-	value	+1	MT start

1 Given in PE clock cycles

2 Indicates internal PE event not directly related to FlexRay bus timing

3.3.2.6 Message Buffer Data Size Register (MBDSR)

0x000C Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MBSEG2DS							0	MBSEG1DS						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-5. Message Buffer Data Size Register (MBDSR)

This register defines the size of the message buffer data section for the two message buffer segments in a number of two-byte entities.

The FlexRay module provides two independent segments for the individual message buffers. All individual message buffers within one segment have to have the same size for the message buffer data section. This size can be different for the two message buffer segments.

Table 3-13. MBDSR Field Descriptions

Field	Description
14–8 MBSEG2DS	**Message Buffer Segment 2 Data Size** — The field defines the size of the message buffer data section in two-byte entities for message buffers within the *second* message buffer segment.
6–0 MBSEG1DS	**Message Buffer Segment 1 Data Size** — The field defines the size of the message buffer data section in two-byte entities for message buffers within the *first* message buffer segment.

3.3.2.7 Message Buffer Segment Size and Utilization Register (MBSSUTR)

0x000E — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	LAST_MB_SEG1							0	LAST_MB_UTIL						
W																
Reset	0	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1

Figure 3-6. Message Buffer Segment Size and Utilization Register (MBSSUTR)

This register is used to define the last individual message buffer that belongs to the first message buffer segment and the number of the last used individual message buffer.

Table 3-14. MBSSUTR Field Descriptions

Field	Description
14–8 LAST_MB_SEG1	**Last Message Buffer In Segment 1** — This field defines the message buffer number of the last individual message buffer that is assigned to the *first* message buffer segment. The individual message buffers in the *first* segment correspond to the message buffer control registers MBCCSRn, MBCCFRn, MBFIDRn, and MBIDXRn with n less than or equaling LAST_MB_SEG1. The first message buffer segment contains LAST_MB_SEG1+1 individual message buffers. **Note:** The *first* message buffer segment contains *at least* one individual message buffer. The individual message buffers in the *second* message buffer segment correspond to the message buffer control registers MBCCSRn, MBCCFRn, MBFIDRn, MBIDXRn with LAST_MB_SEG1 < n < 128. **Note:** If LAST_MB_SEG1 equals 127 all individual message buffers belong to the *first* message buffer segment and the *second* message buffer segment is empty.
6–0 LAST_MB_UTIL	**Last Message Buffer Utilized** — This field defines the message buffer number of last utilized individual message buffer. The message buffer search engine examines all individual message buffer with a message buffer number n less than or equaling LAST_MB_UTIL. **Note:** If LAST_MB_UTIL equals LAST_MB_SEG1 all individual message buffers belong to the *first* message buffer segment and the *second* message buffer segment is empty.

3.3.2.8 Protocol Operation Control Register (POCR)

0x0014 Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	EOC_AP		ERC_AP		BSY	0	0	0	POCCMD			
W	WME								WMC							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-7. Protocol Operation Control Register (POCR)

The application uses this register to issue

- protocol control commands
- external clock correction commands

Protocol control commands are issued by writing to the POCCMD field. For more information on protocol control commands, see Section 3.7.2, “Protocol Control Command Execution”.

External clock correction commands are issued by writing to the EOC_AP and ERC_AP fields. For more information on external clock correction, refer to Section 3.4.11, “External Clock Synchronization”.

Table 3-15. POCR Field Descriptions

Field	Description
15 WME	**Write Mode External Correction** — This bit controls the write mode of the EOC_AP and ERC_AP fields. 0 Write to EOC_AP and ERC_AP fields on register write. 1 No write to EOC_AP and ERC_AP fields on register write.
11–10 EOC_AP	**External Offset Correction Application** — This field is used to trigger the application of the external offset correction value defined in the Protocol Configuration Register 29 (PCR29). 00 do not apply external offset correction value 01 reserved 10 subtract external offset correction value 11 add external offset correction value
9–8 ERC_AP	**External Rate Correction Application** — This field is used to trigger application of the external rate correction value defined in the Protocol Configuration Register 21 (PCR21) 00 do not apply external rate correction value 01 reserved 10 subtract external rate correction value 11 add external rate correction value

Table 3-15. POCR Field Descriptions (Continued)

Field	Description
7 BSY	**Protocol Control Command Write Busy** — This status bit indicates the acceptance of the protocol control command issued by the application via the POCCMD field. The FlexRay module sets this status bit when the application has issued a protocol control command via the POCCMD field. The FlexRay module clears this status bit when protocol control command was accepted by the PE.When the application issues a protocol control command while the BSY bit is asserted, the FlexRay module ignores this command, sets the protocol command ignored error flag PCMI_EF in the CHI Error Flag Register (CHIERFR), and does not change the value of the POCCMD field. 0 Command write idle, command accepted and ready to receive new protocol command. 1 Command write busy, command not yet accepted, not ready to receive new protocol command.
WMC	**Write Mode Command** — This bit controls the write mode of the POCCMD field. 0 Write to POCCMD field on register write. 1 Do not write to POCCMD field on register write.
3–0 POCCMD	**Protocol Control Command** — The application writes to this field to issue a protocol control command to the PE. The FlexRay module sends the protocol command to the PE immediately. While the transfer is running, the BSY bit is set. 0000 ALLOW_COLDSTART — Immediately activate capability of node to cold start cluster. 0001 ALL_SLOTS — Delayed[1] transition to the all slots transmission mode. 0010 CONFIG — Immediately transition to the *POC:config* state. 0011 FREEZE — Immediately transition to the *POC:halt* state. 0100 READY, CONFIG_COMPLETE — Immediately transition to the *POC:ready* state. 0101 RUN — Immediately transition to the *POC:startup start* state. 0110 DEFAULT_CONFIG — Immediately transition to the *POC:default config* state. 0111 HALT — Delayed transition to the *POC:halt* state 1000 WAKEUP — Immediately initiate the wakeup procedure. 1001 reserved 1010 reserved 1011 reserved 1100 RESET[2] — Immediately reset the Protocol Engine. 1101 reserved 1110 reserved 1111 reserved

[1] Delayed means on completion of current communication cycle.

[2] Additional to *FlexRay Communications System Protocol Specification, Version 2.1 Rev A*

NOTE

After sending the RESET command, it is mandatory to execute the command sequence described in Section 3.7.3, “Protocol Reset Command” immediately, to reach the DEFAULT CONFIG state correctly.

3.3.2.9 Global Interrupt Flag and Enable Register (GIFER)

0x0016 Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MIF	PRIF	CHIF	WUPIF	FNEBIF	FNEAIF	RBIF	TBIF	MIE	PRIE	CHIE	WUPIE	FNEBIE	FNEAIE	RBIE	TBIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-8. Global Interrupt Flag and Enable Register (GIFER)

This register provides the means to control some of the interrupt request lines and provides the corresponding interrupt flags. The interrupt flags MIF, PRIF, CHIF, RBIF, and TBIF are the outcome of a binary OR of the related individual interrupt flags and interrupt enables. The generation scheme for these flags is depicted in Figure 3-140. For more details on interrupt generation, see Section 3.4.19, “Interrupt Support. These flags are cleared automatically when all of the corresponding interrupt flags or interrupt enables in the related interrupt flag and enable registers are cleared by the application. In this register the application can clear only the interrupt flags WUPIF, FNEBIF, and FNEAIF, by writing 1 to each them. Writing 0 does not change the flag state. If the application clears a flag and the FlexRay module sets the flag on the same cycle, then that flag remains set.

Table 3-16. GIFER Field Descriptions (Sheet 1 of 2)

Field	Description
15 MIF	**Module Interrupt Flag** — This flag is set if at least one of the other interrupt flags is in this register is asserted and the related interrupt enable is asserted, too. The FlexRay module generates the module interrupt request if MIE is asserted. 0 No interrupt flag is asserted or no interrupt enable is set 1 At least one of the other interrupt flags in this register is asserted and the related interrupt bit is asserted, too
13 PRIF	**Protocol Interrupt Flag** — This flag is set if at least one of the individual protocol interrupt flags in the Protocol Interrupt Flag Register 0 (PIFR0) and Protocol Interrupt Flag Register 1 (PIFR1) is asserted and the related interrupt enable flag is asserted, too. The FlexRay module generates the combined protocol interrupt request if the PRIE flag is asserted. 0 All individual protocol interrupt flags are equal to 0 or no interrupt enable bit is set. 1 At least one of the individual protocol interrupt flags and the related interrupt enable is equal to 1.
13 CHIF	**CHI Interrupt Flag** — This flag is set if at least one of the individual CHI error flags in the CHI Error Flag Register (CHIERFR) is asserted and the chi error interrupt enable GIFER.CHIE is asserted. The FlexRay module generates the combined CHI error interrupt if the CHIE flag is asserted, too. 0 All CHI error flags are equal to 0 or the chi error interrupt is disabled 1 At least one CHI error flag is asserted and chi error interrupt is enabled
12 WUPIF	**Wakeup Interrupt Flag** — This flag is set when the FlexRay module has received a wakeup symbol on the FlexRay bus. The application can determine on which channel the wakeup symbol was received by reading the related wakeup flags WUB and WUA in the Protocol Status Register 3 (PSR3). The FlexRay module generates the wakeup interrupt request if the WUPIE flag is asserted. 0 No wakeup condition or interrupt disabled 1 Wakeup symbol received on FlexRay bus and interrupt enabled
11 FNEBIF	**Receive FIFO channel B Not Empty Interrupt Flag** — This flag is set when the receive FIFO for channel B is not empty. If the application writes 1 to this bit, the FlexRay module updates the FIFO status, increments or wraps the FIFO read index in the Receive FIFO B Read Index Register (RFBRIR) and clears the interrupt flag if the FIFO B is now empty. If the FIFO remains not empty, the FlexRay module sets this flag again. The FlexRay module generates the Receive FIFO B Not empty interrupt if the FNEBIE flag is asserted. 0 Receive FIFO B is empty or interrupt is disabled 1 Receive FIFO B is not empty and interrupt enabled
10 FNEAIF	**Receive FIFO channel A Not Empty Interrupt Flag** — This flag is set when the receive FIFO for channel A is not empty. If the application writes 1 to this bit, the FlexRay module updates the FIFO status, increments or wraps the FIFO read index in the Receive FIFO A Read Index Register (RFARIR) and clears the interrupt flag if the FIFO A is now empty. If the FIFO remains not empty, the FlexRay module sets this flag again. The FlexRay module generates the Receive FIFO A Not empty interrupt if the FNEAIE flag is asserted. 0 Receive FIFO A is empty or interrupt is disabled 1 Receive FIFO A is not empty and interrupt enabled

Table 3-16. GIFER Field Descriptions (Sheet 2 of 2)

Field	Description
9 RBIF	**Receive Message Buffer Interrupt Flag** — This flag is set if for at least one of the individual receive message buffers (MBCCSn.MTD = 0) both the interrupt flag MBIF and the interrupt enable bit MBIE in the corresponding Message Buffer Configuration, Control, Status Registers (MBCCSRn) are asserted. The application can not clear this RBIF flag directly. This flag is cleared by the FlexRay module when all of the interrupt flags MBIF of the individual receive message buffers are cleared by the application or if the application has cleared the interrupt enables bit MBIE. 0 None of the individual receive message buffers has the MBIF and MBIE flag asserted. 1 At least one individual receive message buffer has the MBIF and MBIE flag asserted.
8 TBIF	**Transmit Buffer Interrupt Flag** — This flag is set if for at least one of the individual single or double transmit message buffers (MBCCSn.MTD = 0) both the interrupt flag MBIF and the interrupt enable bit MBIE in the corresponding Message Buffer Configuration, Control, Status Registers (MBCCSRn) are equal to 1. The application can not clear this TBIF flag directly. This flag is cleared by the FlexRay module when all of the individual interrupt flags MBIF of the individual transmit message buffers are cleared by the application or the host has cleared the interrupt enables bit MBIE. 0 None of the individual transmit message buffers has the MBIF and MBIE flag asserted. 1 At least one individual transmit message buffer has the MBIF and MBIE flag asserted.
7 MIE	**Module Interrupt Enable** — This flag controls if the module interrupt line is asserted when the MIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
6 PRIE	**Protocol Interrupt Enable** — This flag controls if the protocol interrupt line is asserted when the PRIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
5 CHIE	**CHI Interrupt Enable** — This flag controls if the CHI interrupt line is asserted when the CHIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
4 WUPIE	**Wakeup Interrupt Enable** — This flag controls if the wakeup interrupt line is asserted when the WUPIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
3 FNEBIE	**Receive FIFO channel B Not Empty Interrupt Enable** — This flag controls if the receive FIFO B interrupt line is asserted when the FNEBIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
2 FNEAIE	**Receive FIFO channel A Not Empty Interrupt Enable** — This flag controls if the receive FIFO A interrupt line is asserted when the FNEAIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
1 RBIE	**Receive Buffer Interrupt Enable** — This flag controls if the receive buffer interrupt line is asserted when the RBIF flag is set. 0 Disable interrupt line 1 Enable interrupt line
0 TBIE	**Transmit Interrupt Enable** — This flag controls if the transmit buffer interrupt line is asserted when the TBIF flag is set. 0 Disable interrupt line 1 Enable interrupt line

3.3.2.10 Protocol Interrupt Flag Register 0 (PIFR0)

0x0018 Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FATL_IF	INTL_IF	ILCF_IF	CSA_IF	MRC_IF	MOC_IF	CCL_IF	MXS_IF	MTX_IF	LTXB_IF	LTXA_IF	TBVB_IF	TBVA_IF	TI2_IF	TI1_IF	CYS_IF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-9. Protocol Interrupt Flag Register 0 (PIFR0)

The register holds one set of the protocol-related individual interrupt flags. The application can clear each interrupt flag by writing a 1 to it. Writing a 0 does not change the state of the flag..

Table 3-17. PIFR0 Field Descriptions

Field	Description
15 FATL_IF	**Fatal Protocol Error Interrupt Flag** — This flag is set when the protocol engine has detected a fatal protocol error. In this case, the protocol engine goes into the *POC:halt* state immediately. The fatal protocol errors are: 1) *pLatestTx* violation, as described in the MAC process of the FlexRay protocol 2) transmission across slot boundary violation, as described in the FSP process of the FlexRay protocol 0 No such event. 1 Fatal protocol error detected.
14 INTL_IF	**Internal Protocol Error Interrupt Flag** — This flag is set when the protocol engine has detected an internal protocol error. In this case, the protocol engine goes into the *POC:halt* state immediately. An internal protocol error occurs when the protocol engine has not finished a calculation and a new calculation is requested. This can be caused by a hardware error. 0 No such event. 1 Internal protocol error detected.
13 ILCF_IF	**Illegal Protocol Configuration Interrupt Flag** — This flag is set when the protocol engine has detected an illegal protocol configuration parameter setting. In this case, the protocol engine goes into the *POC:halt* state immediately. The protocol engine checks the *listen_timeout* value programmed into the Protocol Configuration Register 14 (PCR14) and Protocol Configuration Register 15 (PCR15) when the CONFIG_COMPLETE command was sent by the application via the Protocol Operation Control Register (POCR). If the value of *listen_timeout* is equal to zero, the protocol configuration setting is considered as illegal. 0 No such event. 1 Illegal protocol configuration detected.
12 CSA_IF	**Cold Start Abort Interrupt Flag** — This flag is set when the configured number of allowed cold start attempts is reached and none of these attempts was successful. The number of allowed cold start attempts is configured by the coldstart_attempts field in the Protocol Configuration Register 3 (PCR3). 0 No such event. 1 Cold start aborted and no more coldstart attempts allowed.
11 MRC_IF	**Missing Rate Correction Interrupt Flag** — This flag is set when an insufficient number of measurements is available for rate correction at the end of the communication cycle. 0 No such event 1 Insufficient number of measurements for rate correction detected
10 MOC_IF	**Missing Offset Correction Interrupt Flag** — This flag is set when an insufficient number of measurements is available for offset correction. This is related to the MISSING_TERM event in the CSP process for offset correction in the FlexRay protocol. 0 No such event. 1 Insufficient number of measurements for offset correction detected.

Table 3-17. PIFR0 Field Descriptions (Continued)

Field	Description
9 CCL_IF	**Clock Correction Limit Reached Interrupt Flag** — This flag is set when the internal calculated offset or rate calculation values have reached or exceeded its configured thresholds as given by the *offset_coorection_out* field in the Protocol Configuration Register 9 (PCR9) and the *rate_correction_out* field in the Protocol Configuration Register 14 (PCR14). 0 No such event. 1 Offset or rate correction limit reached.
8 MXS_IF	**Max Sync Frames Detected Interrupt Flag** — This flag is set when the number of synchronization frames detected in the current communication cycle exceeds the value of the *node_sync_max* field in the Protocol Configuration Register 30 (PCR30). 0 No such event. 1 More than *node_sync_max* sync frames detected. **Note:** Only synchronization frames that have passed the synchronization frame acceptance and rejection filters are taken into account.
7 MTX_IF	**Media Access Test Symbol Received Interrupt Flag** — This flag is set when the MTS symbol was received on channel A or channel B. 0 No such event. 1 MTS symbol received.
6 LTXB_IF	*pLatestTx* **Violation on Channel B Interrupt Flag** — This flag is set when the frame transmission on channel B in the dynamic segment exceeds the dynamic segment boundary. This is related to the *pLatestTx* violation, as described in the MAC process of the FlexRay protocol. 0 No such event. 1 *pLatestTx* violation occurred on channel B.
5 LTXA_IF	*pLatestTx* **Violation on Channel A Interrupt Flag** — This flag is set when the frame transmission on channel A in the dynamic segment exceeds the dynamic segment boundary. This is related to the *pLatestTx* violation as described in the MAC process of the FlexRay protocol. 0 No such event. 1 *pLatestTx* violation occurred on channel A.
4 TBVB_IF	**Transmission across boundary on channel B Interrupt Flag** — This flag is set when the frame transmission on channel B crosses the slot boundary. This is related to the transmission across slot boundary violation as described in the FSP process of the FlexRay protocol. 0 No such event. 1 Transmission across boundary violation occurred on channel B.
3 TBVA_IF	**Transmission across boundary on channel A Interrupt Flag** — This flag is set when the frame transmission on channel A crosses the slot boundary. This is related to the transmission across slot boundary violation as described in the FSP process of the FlexRay protocol. 0 No such event. 1 Transmission across boundary violation occurred on channel A.
2 TI2_IF	**Timer 2 Expired Interrupt Flag** — This flag is set when timer 2 expires. 0 No such event. 1 Timer 2 has reached its time limit.
1 TI1_IF	**Timer 1 Expired Interrupt Flag** — This flag is set when timer 1 expires. 0 No such event 1 Timer 1 has reached its time limit
0 CYS_IF	**Cycle Start Interrupt Flag** — This flag is set when a communication cycle starts. 0 No such event 1 Communication cycle started.

3.3.2.11 Protocol Interrupt Flag Register 1 (PIFR1)

0x001A — Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EMC_IF	IPC_IF	PECF_IF	PSC_IF	SSI3_IF	SSI2_IF	SSI1_IF	SSI0_IF	0	0	EVT_IF	ODT_IF	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-10. Protocol Interrupt Flag Register 1 (PIFR1)

The register holds one set of the protocol-related individual interrupt flags. The application can clear each interrupt flag by writing a 1 to it. Writing a 0 does not change the state of the flag.

Table 3-18. PIFR1 Field Descriptions

Field	Description
15 EMC_IF	**Error Mode Changed Interrupt Flag** — This flag is set when the value of the ERRMODE bit field in the Protocol Status Register 0 (PSR0) is changed by the FlexRay module. 0 No such event. 1 ERRMODE field changed.
14 IPC_IF	**Illegal Protocol Control Command Interrupt Flag** — This flag is set when the PE tries to execute a protocol control command, which was issued via the POCCMD field of the Protocol Operation Control Register (POCR), and detects that this protocol control command is not allowed in the current protocol state. In this case the command is not executed. For more details, see Section 3.7.2, “Protocol Control Command Execution”. 0 No such event. 1 Illegal protocol control command detected.
13 PECF_IF	**Protocol Engine Communication Failure Interrupt Flag** — This flag is set if the FlexRay module has detected a communication failure between the protocol engine and the controller host interface 0 No such event. 1 Protocol Engine Communication Failure detected.
12 PSC_IF	**Protocol State Changed Interrupt Flag** — This flag is set when the protocol state in the PROTSTATE field in the Protocol Status Register 0 (PSR0) has changed. 0 No such event. 1 Protocol state changed.
11–8 SSI[3:0]_IF	**Slot Status Counter Incremented Interrupt Flag** — Each of these flags is set when the SLOTSTATUSCNT field in the corresponding Slot Status Counter Registers (SSCR0–SSCR3) is incremented. 0 No such event. 1 The corresponding slot status counter has incremented.
5 EVT_IF	**Even Cycle Table Written Interrupt Flag** — This flag is set if the FlexRay module has written the sync frame measurement / ID tables into the FlexRay Memory for the even cycle. 0 No such event. 1 Sync frame measurement table written
4 ODT_IF	**Odd Cycle Table Written Interrupt Flag** — This flag is set if the FlexRay module has written the sync frame measurement / ID tables into the FlexRay Memory for the odd cycle. 0 No such event. 1 Sync frame measurement table written

3.3.2.12 Protocol Interrupt Enable Register 0 (PIER0)

0x001C — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FATL_IE	INTL_IE	ILCF_IE	CSA_IE	MRC_IE	MOC_IE	CCL_IE	MXS_IE	MTX_IE	LTXB_IE	LTXA_IE	TBVB_IE	TBVA_IE	TI2_IE	TI1_IE	CYS_IE
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-11. Protocol Interrupt Enable Register 0 (PIER0)

This register defines whether or not the individual interrupt flags defined in the Protocol Interrupt Flag Register 0 (PIFR0) can generate a protocol interrupt request.

Table 3-19. PIER0 Field Descriptions

Field	Description
15 FATL_IE	**Fatal Protocol Error Interrupt Enable** — This bit controls FATL_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
14 INTL_IE	**Internal Protocol Error Interrupt Enable** — This bit controls INTL_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
13 ILCF_IE	**Illegal Protocol Configuration Interrupt Enable** — This bit controls ILCF_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
12 CSA_IE	**Cold Start Abort Interrupt Enable** — This bit controls CSA_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
11 MRC_IE	**Missing Rate Correction Interrupt Enable** — This bit controls MRC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
10 MOC_IE	**Missing Offset Correction Interrupt Enable** — This bit controls MOC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
9 CCL_IE	**Clock Correction Limit Reached Interrupt Enable** — This bit controls CCL_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
8 MXS_IE	**Max Sync Frames Detected Interrupt Enable** — This bit controls MXS_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
7 MTX_IE	**Media Access Test Symbol Received Interrupt Enable** — This bit controls MTX_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
6 LTXB_IE	*pLatestTx* **Violation on Channel B Interrupt Enable** — This bit controls LTXB_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
5 LTXA_IE	*pLatestTx* **Violation on Channel A Interrupt Enable** — This bit controls LTXA_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled

Table 3-19. PIER0 Field Descriptions (Continued)

Field	Description
4 TBVB_IE	**Transmission across boundary on channel B Interrupt Enable** — This bit controls TBVB_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
3 TBVA_IE	**Transmission across boundary on channel A Interrupt Enable** — This bit controls TBVA_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
2 TI2_IE	**Timer 2 Expired Interrupt Enable** — This bit controls TI1_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
1 TI1_IE	**Timer 1 Expired Interrupt Enable** — This bit controls TI1_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
0 CYS_IE	**Cycle Start Interrupt Enable** — This bit controls CYC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled

3.3.2.13 Protocol Interrupt Enable Register 1 (PIER1)

0x001E — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EMC_IE	IPC_IE	PECF_IE	PSC_IE	SSI3_IE	SSI2_IE	SSI1_IE	SSI0_IE	0	0	EVT_IE	ODT_IE	0	0	0	0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-12. Protocol Interrupt Enable Register 1 (PIER1)

This register defines whether or not the individual interrupt flags defined in Protocol Interrupt Flag Register 1 (PIFR1) can generate a protocol interrupt request.

Table 3-20. PIER1 Field Descriptions

Field	Description
15 EMC_IE	**Error Mode Changed Interrupt Enable** — This bit controls EMC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
14 IPC_IE	**Illegal Protocol Control Command Interrupt Enable** — This bit controls IPC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
13 PECF_IE	**Protocol Engine Communication Failure Interrupt Enable** — This bit controls PECF_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
12 PSC_IE	**Protocol State Changed Interrupt Enable** — This bit controls PSC_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled

Table 3-20. PIER1 Field Descriptions (Continued)

Field	Description
11–8 SSI[3:0]_IE	**Slot Status Counter Incremented Interrupt Enable** — This bit controls SSI[3:0]_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
5 EVT_IE	**Even Cycle Table Written Interrupt Enable** — This bit controls EVT_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled
4 ODT_IE	**Odd Cycle Table Written Interrupt Enable** — This bit controls ODT_IF interrupt request generation. 0 interrupt request generation disabled 1 interrupt request generation enabled

3.3.2.14 CHI Error Flag Register (CHIERFR)

0x0020 Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FRLB_EF	FRLA_EF	PCMI_EF	FOVB_EF	FOVA_EF	MBS_EF	MBU_EF	LCK_EF	DBL_EF	SBCF_EF	FID_EF	DPL_EF	SPL_EF	NML_EF	NMF_EF	ILSA_EF
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-13. CHI Error Flag Register (CHIERFR)

This register holds the CHI related error flags. The application can clear each error flag by writing a 1 to it. Writing a 0 does not change the state of the flag. The interrupt generation for each of these error flags is controlled by the CHI interrupt enable bit CHIE in the Global Interrupt Flag and Enable Register (GIFER).

Table 3-21. CHIERFR Field Descriptions

Field	Description
15 FRLB_EF	**Frame Lost Channel B Error Flag** — This flag is set if a complete frame was received on channel B but could not be stored in the selected individual message buffer because this message buffer is currently locked by the application. In this case, the frame and the related slot status information are lost. 0 No such event 1 Frame lost on channel B detected
14 FRLA_EF	**Frame Lost Channel A Error Flag** — This flag is set if a complete frame was received on channel A but could not be stored in the selected individual message buffer because this message buffer is currently locked by the application. In this case, the frame and the related slot status information are lost. 0 No such error 1 Frame lost on channel A detected
13 PCMI_EF	**Protocol Command Ignored Error Flag** — This flag is set if the application has issued a POC command by writing to the POCCMD field in the Protocol Operation Control Register (POCR) while the BSY flag is equal to 1. In this case the command is ignored by the FlexRay module and is lost. 0 No such error 1 POC command ignored

Table 3-21. CHIERFR Field Descriptions (Continued)

Field	Description
12 FOVB_EF	**Receive FIFO Overrun Channel B Error Flag** — This flag is set when an overrun of the Receive FIFO for channel B occurred. This error occurs if a semantically valid frame was received on channel B and matches the all criteria to be appended to the FIFO for channel B but the FIFO is full. In this case, the received frame and its related slot status information is lost. 0 No such error 1 Receive FIFO overrun on channel B has been detected
11 FOVA_EF	**Receive FIFO Overrun Channel A Error Flag** — This flag is set when an overrun of the Receive FIFO for channel A occurred. This error occurs if a semantically valid frame was received on channel A and matches the all criteria to be appended to the FIFO for channel A but the FIFO is full. In this case, the received frame and its related slot status information is lost. 0 No such error 1 Receive FIFO overrun on channel B has been detected
10 MSB_EF	**Message Buffer Search Error Flag** — This flag is set if the message buffer search engine continues running while the next search cycle must be started due to the FlexRay protocol timing. In this case, not all message buffers are considered while searching. 0 No such event 1 Search engine active while search start appears
9 MBU_EF	**Message Buffer Utilization Error Flag** — This flag is asserted if the application writes to a message buffer control field that is beyond the number of utilized message buffers programmed in the **Message Buffer Segment Size and Utilization Register (MBSSUTR).** If the application writes to a MBCCSRn register with n > LAST_MB_UTIL, the FlexRay module ignores the write attempt and asserts the message buffer utilization error flag MBU_EF in the CHI Error Flag Register (CHIERFR). 0 No such event 1 Non-utilized message buffer enabled
8 LCK_EF	**Lock Error Flag** — This flag is set if the application tries to lock a message buffer that is already locked by the FlexRay module due to internal operations. In that case, the FlexRay module does not grant the lock to the application. The application must issue the lock request again. 0 No such error 1 Lock error detected
7 DBL_EF	**Double Transmit Message Buffer Lock Error Flag** — This flag is set if the application tries to lock the transmit side of a double transmit message buffer. In this case, the FlexRay module does not grant the lock to the transmit side of a double transmit message buffer. 0 No such event 1 Double transmit buffer lock error occurred
6 SBCF_EF	**System Bus Communication Failure Error Flag** — This flag is set if the FlexRay module was not able to transmit or receive data via the system bus in time. In the case of writing, data is lost; in the case of reading, the transmission onto the FlexRay bus is stopped for the current slot and resumed in the next slot. 0 No such event 1 System bus communication failure occurred
5 FID_EF	**Frame ID Error Flag** — This flag is set if the frame ID stored in the message buffer header area differs from the frame ID stored in the message buffer control register. 0 No such error occurred 1 Frame ID error occurred
4 DPL_EF	**Dynamic Payload Length Error Flag** — This flag is set if the payload length written into the message buffer header field of a single or double transmit message buffer assigned to the dynamic segment is greater than the maximum payload length for the dynamic segment as it is configured in the corresponding protocol configuration register field max_payload_length_dynamic in the Protocol Configuration Register 24 (PCR24). 0 No such error occurred 1 Dynamic payload length error occurred

Table 3-21. CHIERFR Field Descriptions (Continued)

Field	Description
3 SPL_EF	**Static Payload Length Error** Flag — This flag is set if the payload length written into the message buffer header field of a single or double transmit message buffer assigned to the static segment is different from the payload length for the static segment as it is configured in the corresponding protocol configuration register field payload_length_static in the Protocol Configuration Register 19 (PCR19). 0 No such error occurred 1 Static payload length error occurred
2 NML_EF	**Network Management Length Error Flag** — This flag is set if the payload length written into the header structure of a receive message buffer assigned to the static segment is less than the configured length of the Network Management Vector as configured in the Network Management Vector Length Register (NMVLR). In this case, the received part of the Network Management Vector is used to update the Network Management Vector. 0 No such error occurred 1 Network management length error occurred
1 NMF_EF	**Network Management Frame Error Flag** — This flag is set if a received message in the static segment with a Preamble Indicator flag PP asserted has its Null Frame indicator flag NF asserted as well. In this case, the Global Network Management Registers (see Network Management Vector Registers (NMVR0–NMVR5)) are not updated. 0 No such error occurred 1 Network management frame error occurred
0 ILSA_EF	**Illegal System Memory Access Error Flag** — This flag is set if the external system memory subsystem has detected and indicated an illegal system memory access from the FlexRay module. The exact meaning of an illegal system memory access is defined by the current implementation of the memory subsystem. 0 No such event. 1 Illegal system memory access occurred.

3.3.2.15 Message Buffer Interrupt Vector Register (MBIVEC)

0x0022

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	TBIVEC							0	RBIVEC						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-14. Message Buffer Interrupt Vector Register (MBIVEC)

This register indicates the lowest numbered receive message buffer and the lowest numbered transmit message buffer that have their interrupt status flag MBIF and interrupt enable MBIE bits asserted. This means that message buffers with lower message buffer numbers have higher priority.

Table 3-22. MBIVEC Field Descriptions

Field	Description
14-8 TBIVEC	**Transmit Buffer Interrupt Vector** — This field provides the number of the lowest numbered enabled transmit message buffer that has its interrupt status flag MBIF and its interrupt enable bit MBIE set. If there is no transmit message buffer with the interrupt status flag MBIF and the interrupt enable MBIE bits asserted, the value in this field is set to 0.
6-0 RBIVEC	**Receive Buffer Interrupt Vector** — This field provides the message buffer number of the lowest numbered receive message buffer which has its interrupt flag MBIF and its interrupt enable bit MBIE asserted. If there is no receive message buffer with the interrupt status flag MBIF and the interrupt enable MBIE bits asserted, the value in this field is set to 0.

3.3.2.16 Channel A Status Error Counter Register (CASERCR)

0x0024 Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS_ERR_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-15. Channel A Status Error Counter Register (CASERCR)

This register provides the channel status error counter for channel A. The protocol engine generates a slot status vector for each static slot, each dynamic slot, the symbol window, and the NIT. The slot status vector contains the four protocol related error indicator bits *vSS!SyntaxError, vSS!ContentError, vSS!BViolation*, and *vSS!TxConflict*. The FlexRay module increments the status error counter by 1 if, for a slot or segment, at least one error indicator bit is set to 1. The counter wraps around after it has reached the maximum value. For more information on slot status monitoring, see Section 3.4.18, “Slot Status Monitoring”.

Table 3-23. CASERCR Field Descriptions

Field	Description
15–0 STATUS_ERR_CNT	**Channel Status Error Counter** — This field provides the current value channel status error counter. The counter value is updated within the first macrotick of the following slot or segment.

3.3.2.17 Channel B Status Error Counter Register (CBSERCR)

0x0026 Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS_ERR_CNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-16. Channel B Status Error Counter Register (CBSERCR)

This register provides the channel status error counter for channel B. The protocol engine generates a slot status vector for each static slot, each dynamic slot, the symbol window, and the NIT. The slot status vector contains the four protocol related error indicator bits *vSS!SyntaxError*, *vSS!ContentError*, *vSS!BViolation*, and *vSS!TxConflict*. The FlexRay module increments the status error counter by 1 if, for a slot or segment, at least one error indicator bit is set to 1. The counter wraps around after it has reached the maximum value. For more information on slot status monitoring see Section 3.4.18, “Slot Status Monitoring”.

Table 3-24. CBSERCR Field Descriptions

Field	Description
15–0 STATUS_ERR_CNT	**Channel Status Error Counter** — This field provides the current channel status error count. The counter value is updated within the first macrotick of the following slot or segment.

3.3.2.18 Protocol Status Register 0 (PSR0)

0x0028

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERRMODE		SLOTMODE		0	PROTSTATE			STARTUPSTATE				0	WAKEUPSTATUS		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-17. Protocol Status Register 0 (PSR0)

This register provides information about the current protocol status.

Table 3-25. PSR0 Field Descriptions (Sheet 1 of 2)

Field	Description
15–14 ERRMODE	**Error Mode** — protocol related variable: *vPOC!ErrorMode*. This field indicates the error mode of the protocol. 00 ACTIVE 01 PASSIVE 10 COMM_HALT 11 reserved
13–12 SLOTMODE	**Slot Mode** — protocol related variable: *vPOC!SlotMode*. This field indicates the slot mode of the protocol. 00 SINGLE 01 ALL_PENDING 10 ALL 11 reserved
10–8 PROTSTATE	**Protocol State** — protocol related variable: *vPOC!State.* This field indicates the state of the protocol. 000 *POC:default config* 001 *POC:config* 010 *POC:wakeup* 011 *POC:ready* 100 *POC:normal passive* 101 *POC:normal active* 110 *POC:halt* 111 *POC:startup*

Table 3-25. PSR0 Field Descriptions (Sheet 2 of 2)

Field	Description
7–4 STARTUP STATE	**Startup State** — protocol related variable: *vPOC!StartupState.* This field indicates the current sub-state of the startup procedure. 0000 reserved 0001 reserved 0010 *POC:coldstart collision resolution* 0011 *POC:coldstart listen* 0100 *POC:integration consistency check* 0101 *POC:integrationi listen* 0110 reserved 0111 *POC:initialize schedule* 1000 reserved 1001 reserved 1010 *POC:coldstart consistency check* 1011 reserved 1100 reserved 1101 *POC:integration coldstart check* 1110 *POC:coldstart gap* 1111 *POC:coldstart join*
2–0 WAKEUP STATUS	**Wakeup Status** — protocol related variable: *vPOC!WakeupStatus*. This field provides the outcome of the execution of the wakeup mechanism. 000 UNDEFINED 001 RECEIVED_HEADER 010 RECEIVED_WUP 011 COLLISION_HEADER 100 COLLISION_WUP 101 COLLISION_UNKNOWN 110 TRANSMITTED 111 reserved

3.3.2.19 Protocol Status Register 1 (PSR1)

0x002A — Additional Reset: CSAA, CSP, CPN: RUN Command — Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSAA	CSP	0	REMCSAT					CPN	HHR	FRZ	APTAC				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-18. Protocol Status Register 1 (PSR1)

Table 3-26. PSR1 Field Descriptions

Field	Description
15 CSAA	**Cold Start Attempt Aborted Flag** — Protocol related event: Set coldstart abort indicator in CHI' This flag bit is set when the FlexRay modulehas aborted a cold start attempt. The application clears this flag by writing 1 to it. Writing a 0 does not change the state of the flag. If the application clears the flag while the FlexRay module sets the flag at the same time, then the flag is not cleared. 0 No such event 1 Cold start attempt aborted
14 CSP	**Leading Cold Start Path** — This status bit is set when the FlexRay module has reached the *POC:normal active* state via the leading cold start path. This indicates that this node has started the network 0 No such event 1 *POC:normal active* reached from *POC:startup* state via leading cold start path
12–8 REMCSAT	**Remaining Coldstart Attempts** — protocol related variable: *vRemainingColdstartAttempts* This field provides the number of remaining cold start attempts that the FlexRay module executes.
7 CPN	**Leading Cold Start Path Noise** — protocol related variable: *vPOC!ColdstartNoise* This status bit is set if the FlexRay module has reached the *POC:normal active* state via the leading cold start path under noise conditions. This indicates there was some activity on the FlexRay bus while the FlexRay module was starting up the cluster. 0 No such event 1 *POC:normal active* state was reached from *POC:startup* state via noisy leading cold start path
6 HHR	**Host Halt Request Pending** — protocol related variable: *vPOC!CHIHaltRequest* This status bit is set when FlexRay module receives the HALT command from the application via the Protocol Operation Control Register (POCR). The FlexRay module clears this status bit after a hard reset condition or when the protocol is in the *POC:default config* state. 0 No such event 1 HALT command received
5 FRZ	**Freeze Occurred** — protocol related variable: *vPOC!Freeze* This status bit is set when the FlexRay module has reached the *POC:halt* state due to the host FREEZE command or due to an internal error condition requiring immediate halt. The FlexRay module clears this status bit after a hard reset condition or when the protocol is in the *POC:default config* state. 0 No such event 1 Immediate halt due to FREEZE or internal error condition
4–0 APTAC	**Allow Passive to Active Counter** — protocol related variable: *vPOC!vAllowPassivetoActive* This field provides the number of consecutive even/odd communication cycle pairs that have passed with valid rate and offset correction terms, but the protocol remains in the *POC:normal passive* state due to an application configured delay to enter *POC:normal active* state. This delay is defined by the allow_passive_to_active field in the Protocol Configuration Register 12 (PCR12)..

3.3.2.20 Protocol Status Register 2 (PSR2)

0x002C — Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NBVB	NSEB	STCB	SBVB	SSEB	MTB	NBVA	NSEA	STCA	SBVA	SSEA	MTA	CLKCORRFAILCNT			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-19. Protocol Status Register 2 (PSR2)

This register provides a snapshot of status information about the Network Idle Time NIT, the Symbol Window and the clock synchronization. The NIT related status bits NBVB, NSEB, NBVA, and NSEA are updated by the FlexRay module after the end of the NIT and before the end of the first slot of the next

communication cycle. The Symbol Window related status bits STCB, SBVB, SSEB, MTB, STCA, SBVA, SSEB, and MTA are updated by the FlexRay module after the end of the symbol window and before the end of the current communication cycle. If no symbol window is configured, the symbol window related status bits remain in their reset state. The clock synchronization related CLKCORRFAILCNT is updated by the FlexRay module after the end of the static segment and before the end of the current communication cycle.

Table 3-27. PSR2 Field Descriptions

Field	Description
15 NBVB	**NIT Boundary Violation on Channel B** — protocol related variable: *vSS!BViolation* for NIT on channel B This status bit is set when there was some media activity on the FlexRay bus channel B at the end of the NIT. 0 No such event 1 Media activity at boundaries detected
14 NSEB	**NIT Syntax Error on Channel B** — protocol related variable: *vSS!SyntaxError* for NIT on channel B This status bit is set when a syntax error was detected during NIT on channel B. 0 No such event 1 Syntax error detected
13 STCB	**Symbol Window Transmit Conflict on Channel B** — protocol related variable: *vSS!TxConflict* for symbol window on channel B This status bit is set if there was a transmission conflict during the symbol window on channel B. 0 No such event 1 Transmission conflict detected
12 SBVB	**Symbol Window Boundary Violation on Channel B** — protocol related variable: *vSS!BViolation* for symbol window on channel B This status bit is set if there was some media activity on the FlexRay bus channel B at the start or at the end of the symbol window. 0 No such event 1 Media activity at boundaries detected
11 SSEB	**Symbol Window Syntax Error on Channel B** — protocol related variable: *vSS!SyntaxError* for symbol window on channel B This status bit is set when a syntax error was detected during the symbol window on channel B. 0 No such event 1 Syntax error detected
10 MTB	**Media Access Test Symbol MTS Received on Channel B** — protocol related variable: *vSS!ValidMTS* for Symbol Window on channel B This status bit is set if the Media Access Test Symbol MTS was received in the symbol window on channel B. 0 No such event 1 MTS symbol received
9 NBVA	**NIT Boundary Violation on Channel A** — protocol related variable: *vSS!BViolation* for NIT on channel A This status bit is set when there was some media activity on the FlexRay bus channel A at the end of the NIT. 0 No such event 1 Media activity at boundaries detected
8 NSEA	**NIT Syntax Error on Channel A** — protocol related variable: *vSS!SyntaxError* for NIT on channel A This status bit is set when a syntax error was detected during NIT on channel A. 0 No such event 1 Syntax error detected
7 STCA	**Symbol Window Transmit Conflict on Channel A** — protocol related variable: *vSS!TxConflict* for symbol window on channel A This status bit is set if there was a transmission conflicts during the symbol window on channel A. 0 No such event 1 Transmission conflict detected

Table 3-27. PSR2 Field Descriptions (Continued)

Field	Description
6 SBVA	**Symbol Window Boundary Violation on Channel A** — protocol related variable: *vSS!BViolation* for symbol window on channel A This status bit is set if there was some media activity on the FlexRay bus channel A at the start or at the end of the symbol window. 0 No such event 1 Media activity at boundaries detected
5 SSEA	**Symbol Window Syntax Error on Channel A** — protocol related variable: *vSS!SyntaxError* for symbol window on channel A This status bit is set when a syntax error was detected during the symbol window on channel A. 0 No such event 1 Syntax error detected
4 MTA	**Media Access Test Symbol MTS Received on Channel A** — protocol related variable: *vSS!ValidMTS* for symbol window on channel A This status bit is set if the Media Access Test Symbol MTS was received in the symbol window on channel A. 1 MTS symbol received 0 No such event
3–0 CLKCORR-FAILCNT	**Clock Correction Failed Counter** — protocol related variable: *vClockCorrectionFailed* This field provides the number of consecutive even/odd communication cycle pairs that have passed without clock synchronization having performed an offset or a rate correction due to lack of synchronization frames. It is not incremented when it has reached the configured value of max_without_clock_correction_fatal or max_without_clock_correction_passive as defined in the Protocol Configuration Register 8 (PCR8). The FlexRay module resets this counter on a hard reset condition, when the protocol enters the *POC:normal active* state, or when both the rate and offset correction terms have been calculated successfully.

3.3.2.21 Protocol Status Register 3 (PSR3)

0x002E — Additional Reset: RUN Command — Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	WUB	ABVB	AACB	ACEB	ASEB	AVFB	0	0	WUA	ABVA	AACA	ACEA	ASEA	AVFA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-20. Protocol Status Register 3 (PSR3)

This register provides aggregated channel status information as an accrued status of channel activity for all communication slots, regardless of whether they are assigned for transmission or subscribed for reception. It provides accrued information for the symbol window, the NIT, and the wakeup status. The application can clear any flag at any time by writing a 1 to it. Writing a 0 does not change the flag state. If the application tries to clear a flag while the FlexRay module sets the flag at the same time, then that flag is not cleared.

Table 3-28. PSR3 Field Descriptions

Field	Description
13 WUB	**Wakeup Symbol Received on Channel B** — This flag is set when a wakeup symbol was received on channel B. 0 No wakeup symbol received 1 Wakeup symbol received
12 ABVB	**Aggregated Boundary Violation on Channel B** — This flag is set when a boundary violation has been detected on channel B. Boundary violations are detected in the communication slots, the symbol window, and the NIT. 0 No boundary violation detected 1 Boundary violation detected
11 AACB	**Aggregated Additional Communication on Channel B** — This flag is set when at least one valid frame was received on channel B in a slot that also contained an additional communication with syntax error, content error, or boundary violations. 0 No additional communication detected 1 Additional communication detected
10 ACEB	**Aggregated Content Error on Channel B** — This flag is set when a content error has been detected on channel B. Content errors are detected in the communication slots, the symbol window, and the NIT. 0 No content error detected 1 Content error detected
9 ASEB	**Aggregated Syntax Error on Channel B** — This flag is set when a syntax error has been detected on channel B. Syntax errors are detected in the communication slots, the symbol window and the NIT. 0 No syntax error detected 1 Syntax errors detected
8 AVFB	**Aggregated Valid Frame on Channel B** — This flag is set when a syntactically correct valid frame has been received in any static or dynamic slot through channel B. 1 At least one syntactically valid frame received 0 No syntactically valid frames received
5 WUA	**Wakeup Symbol Received on Channel A** — This flag is set when a wakeup symbol was received on channel A. 0 No wakeup symbol received 1 Wakeup symbol received
4 ABVA	**Aggregated Boundary Violation on Channel A** — This flag is set when a boundary violation has been detected on channel A. Boundary violations are detected in the communication slots, the symbol window, and the NIT. 0 No boundary violation detected 1 Boundary violation detected
3 AACA	**Aggregated Additional Communication on Channel A** — This flag is set when a valid frame was received in a slot on channel A that also contained an additional communication with syntax error, content error, or boundary violations. 0 No additional communication detected 1 Additional communication detected
2 ACEA	**Aggregated Content Error on Channel A** — This flag is set when a content error has been detected on channel A. Content errors are detected in the communication slots, the symbol window, and the NIT. 0 No content error detected 1 Content error detected

Table 3-28. PSR3 Field Descriptions (Continued)

Field	Description
1 ASEA	**Aggregated Syntax Error on Channel A** — This flag is set when a syntax error has been detected on channel A. Syntax errors are detected in the communication slots, the symbol window, and the NIT. 0 No syntax error detected 1 Syntax errors detected
0 AVFA	**Aggregated Valid Frame on Channel A** — This flag is set when a syntactically correct valid frame has been received in any static or dynamic slot through channel A. 0 No syntactically valid frames received 1 At least one syntactically valid frame received

3.3.2.22 Macrotick Counter Register (MTCTR)

0x0030

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	MTCT													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-21. Macrotick Counter Register (MTCTR)

This register provides the macrotick count of the current communication cycle.

Table 3-29. MTCTR Field Descriptions

Field	Description
13–0 MTCT	**Macrotick Counter** — protocol related variable: *vMacrotick* This field provides the macrotick count of the current communication cycle.

3.3.2.23 Cycle Counter Register (CYCTR)

0x0032

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	CYCCNT					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-22. Cycle Counter Register (CYCTR)

This register provides the number of the current communication cycle.

Table 3-30. CYCTR Field Descriptions

Field	Description
5–0 CYCCNT	**Cycle Counter** — protocol related variable: *vCycleCounter* This field provides the number of the current communication cycle. If the counter reaches the maximum value of 63, the counter wraps and starts from zero again.

3.3.2.24 Slot Counter Channel A Register (SLTCTAR)

0x0034

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SLOTCNTA										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-23. Slot Counter Channel A Register (SLTCTAR)

This register provides the number of the current slot in the current communication cycle for channel A.

Table 3-31. SLTCTAR Field Descriptions

Field	Description
10–0 SLOTCNTA	**Slot Counter Value for Channel A** — protocol related variable: *vSlotCounter* for channel A This field provides the number of the current slot in the current communication cycle.

3.3.2.25 Slot Counter Channel B Register (SLTCTBR)

0x0036

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	SLOTCNTB										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-24. Slot Counter Channel B Register (SLTCTBR)

This register provides the number of the current slot in the current communication cycle for channel B.

Table 3-32. SLTCTBR Field Descriptions

Field	Description
10–0 SLOTCNTA	**Slot Counter Value for Channel B** — protocol related variable: *vSlotCounter* for channel B This field provides the number of the current slot in the current communication cycle.

3.3.2.26 Rate Correction Value Register (RTCORVR)

0x0038 Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RATECORR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-25. Rate Correction Value Register (RTCORVR)

This register provides the sign extended rate correction value in microticks as it was calculated by the clock synchronization algorithm. The FlexRay module updates this register during the NIT of each odd numbered communication cycle.

Table 3-33. RTCORVR Field Descriptions

Field	Description
15–0 RATECORR	**Rate Correction Value** — protocol related variable: *vRateCorrection* (before value limitation and external rate correction) This field provides the sign extended rate correction value in microticks as it was calculated by the clock synchronization algorithm. The value is represented in 2's complement format. This value does not include the value limitation and the application of the external rate correction. If the magnitude of the internally calculated rate correction value exceeds the limit given by rate_correction_out in the Protocol Configuration Register 13 (PCR13), the clock correction reached limit interrupt flag CCL_IF is set in the Protocol Interrupt Flag Register 0 (PIFR0). **Note:** If the FlexRay module was not able to calculate a new rate correction term due to a lack of synchronization frames, the RATECORR value is not updated.

3.3.2.27 Offset Correction Value Register (OFCORVR)

0x003A — Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OFFSETCORR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-26. Offset Correction Value Register (OFCORVR)

This register provides the sign extended offset correction value in microticks as it was calculated by the clock synchronization algorithm. The FlexRay module updates this register during the NIT.

Table 3-34. OFCORVR Field Descriptions

Field	Description
15–0 OFFSET- CORR	**Offset Correction Value** — protocol related variable: *vOffsetCorrection* (before value limitation and external offset correction) This field provides the sign extended offset correction value in microticks as it was calculated by the clock synchronization algorithm. The value is represented in 2's complement format. This value does not include the value limitation and the application of the external offset correction. If the magnitude of the internally calculated rate correction value exceeds the limit given by offset_correction_out field in the Protocol Configuration Register 29 (PCR29), the clock correction reached limit interrupt flag CCL_IF is set in the Protocol Interrupt Flag Register 0 (PIFR0). **Note:** If the FlexRay module was not able to calculate an new offset correction term due to a lack of synchronization frames, the OFFSETCORR value is not updated.

3.3.2.28 Combined Interrupt Flag Register (CIFRR)

0x003C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	MIF	PRIF	CHIF	WUPIF	FNEBIF	FNEAIF	RBIF	TBIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-27. Combined Interrupt Flag Register (CIFRR)

This register provides five combined interrupt flags and a copy of three individual interrupt flags. The combined interrupt flags are the result of a binary OR of the values of other interrupt flags regardless of the state of the interrupt enable bits. The generation scheme for the combined interrupt flags is depicted in Figure 3-142. The individual interrupt flags WUPIF, FNEBIF, and FNEAIF are copies of corresponding flags in the Global Interrupt Flag and Enable Register (GIFER) and are provided here to simplify the application interrupt flag check. To clear the individual interrupt flags, the application must use the Global Interrupt Flag and Enable Register (GIFER).

NOTE

The meanings of the five combined status bits MIF, PRIF, CHIF, RBIF, and TBIF are different from those mentioned in the Global Interrupt Flag and Enable Register (GIFER).

Table 3-35. CIFRR Field Descriptions

Field	Description
7 MIF	**Module Interrupt Flag** — This flag is set if there is at least one interrupt source that has its interrupt flag asserted. 0 No interrupt source has its interrupt flag asserted 1 At least one interrupt source has its interrupt flag asserted
6 PRIF	**Protocol Interrupt Flag** — This flag is set if at least one of the individual protocol interrupt flags in the Protocol Interrupt Flag Register 0 (PIFR0) or Protocol Interrupt Flag Register 1 (PIFR1) is equal to 1. 0 All individual protocol interrupt flags are equal to 0 1 At least one of the individual protocol interrupt flags is equal to 1
5 CHIF	**CHI Interrupt Flag** — This flag is set if at least one of the individual CHI error flags in the CHI Error Flag Register (CHIERFR) is equal to 1. 0 All CHI error flags are equal to 0 1 At least one CHI error flag is equal to 1
4 WUPIF	**Wakeup Interrupt Flag** — Provides the same value as GIFER{WUPIF]
3 FNEBIF	**Receive FIFO channel B Not Empty Interrupt Flag** — Provides the same value as GIFER[FNEBIF]
2 FNEAIF	**Receive FIFO channel A Not Empty Interrupt Flag** — Provides the same value as GIFER[FNEAIF]
1 RBIF	**Receive Message Buffer Interrupt Flag** — This flag is set if for at least one of the individual receive message buffers (MBCCSRn[MTD] = 0) the interrupt flag MBIF in the corresponding Message Buffer Configuration, Control, Status Registers (MBCCSRn) is equal to 1. 0 None of the individual receive message buffers has the MBIF flag asserted. 1 At least one individual receive message buffers has the MBIF flag asserted.
0 TBIF	**Transmit Message Buffer Interrupt Flag** — This flag is set if for at least one of the individual single or double transmit message buffers (MBCCSRn[MTD] = 1) the interrupt flag MBIF in the corresponding Message Buffer Configuration, Control, Status Registers (MBCCSRn) is equal to 1. 0 None of the individual transmit message buffers has the MBIF flag asserted. 1 At least one individual transmit message buffers has the MBIF flag asserted.

3.3.2.29 Sync Frame Counter Register (SFCNTR)

0x0040 — Additional Reset: RUN Command

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SFEVB				SFEVA				SFODB				SFODA			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-28. Sync Frame Counter Register (SFCNTR)

This register provides the number of synchronization frames that are used for clock synchronization in the last even and in the last odd numbered communication cycle. This register is updated after the start of the NIT and before 10 MT after offset correction start.

NOTE

If the application has locked the even synchronization table at the end of the static segment of an even communication cycle, the FlexRay module does not update the fields SFEVB and SFEVA.

If the application has locked the odd synchronization table at the end of the static segment of an odd communication cycle, the FlexRay module does not update the values SFODB and SFODA.

Table 3-36. SFCNTR Field Descriptions

Field	Description
15–12 SFEVB	Sync Frames Channel B, even cycle — protocol related variable: size of (*vSyncIdListB* for even cycle) This field provides the size of the internal list of frame IDs of received synchronization frames used for clock synchronization.
11–8 SFEVB	Sync Frames Channel A, even cycle — protocol related variable: size of (*vSyncIdListA* for even cycle) This field provides the size of the internal list of frame IDs of received synchronization frames used for clock synchronization.
7–4 SFODB	Sync Frames Channel B, odd cycle — protocol related variable: size of (*vSyncIdListB* for odd cycle) This field provides the size of the internal list of frame IDs of received synchronization frames used for clock synchronization.
3–0 SFODA	Sync Frames Channel A, odd cycle — protocol related variable: size of (*vSyncIdListA* for odd cycle) This field provides the size of the internal list of frame IDs of received synchronization frames used for clock synchronization.

3.3.2.30 Sync Frame Table Offset Register (SFTOR)

0x0042 — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SFT_OFFSET[15:1]															0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-29. Sync Frame Table Offset Register (SFTOR)

This register defines the Flexray Memory related offset for sync frame tables. For more details, see Section 3.4.12, “Sync Frame ID and Sync Frame Deviation Tables”.

Table 3-37. SFTOR Field Description

Field	Description
15–1 SFTOR	**Sync Frame Table Offset** — The offset of the Sync Frame Tables in the FlexRay Memory. This offset is required to be 16-bit aligned. Thus STF_OFFSET[0] is always 0.

3.3.2.31 Sync Frame Table Configuration, Control, Status Register (SFTCCSR)

0x0044 Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	CYCNUM						ELKS	OLKS	EVAL	OVAL	0	0	SDVEN	SIDEN
W	ELKT	OLKT												OPT		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-30. Sync Frame Table Configuration, Control, Status Register (SFTCCSR)

This register provides configuration, control, and status information related to the generation and access of the clock sync ID tables and clock sync measurement tables. For a detailed description, see Section 3.4.12, “Sync Frame ID and Sync Frame Deviation Tables”.

Table 3-38. SFTCCSR Field Descriptions

Field	Description
15 ELKT	**Even Cycle Tables Lock/Unlock Trigger** — This trigger bit is used to lock and unlock the even cycle tables. 0 No effect 1 Triggers lock/unlock of the even cycle tables.
14 OLKT	**Odd Cycle Tables Lock/Unlock Trigger** — This trigger bit is used to lock and unlock the odd cycle tables. 0 No effect 1 Triggers lock/unlock of the odd cycle tables.
13–8 CYCNUM	**Cycle Number** — This field provides the number of the cycle in which the currently locked table was recorded. If none or both tables are locked, this value is related to the even cycle table.
7 ELKS	**Even Cycle Tables Lock Status** — This status bit indicates whether the application has locked the even cycle tables. 0 Application has not locked the even cycle tables. 1 Application has locked the even cycle tables.
6 OLKS	**Odd Cycle Tables Lock Status** — This status bit indicates whether the application has locked the odd cycle tables. 0 Application has not locked the odd cycle tables. 1 Application has locked the odd cycle tables.
5 EVAL	**Even Cycle Tables Valid** — This status bit indicates whether the Sync Frame ID and Sync Frame Deviation Tables for the even cycle are valid. The FlexRay module clears this status bit when it starts updating the tables, and sets this bit when it has finished the table update. 0 Tables are not valid (update is ongoing) 1 Tables are valid (consistent).
4 OVAL	**Odd Cycle Tables Valid** — This status bit indicates whether the Sync Frame ID and Sync Frame Deviation Tables for the odd cycle are valid. The FlexRay module clears this status bit when it starts updating the tables, and sets this bit when it has finished the table update. 0 Tables are not valid (update is ongoing) 1 Tables are valid (consistent).

Table 3-38. SFTCCSR Field Descriptions (Continued)

Field	Description
2 OPT	**One Pair Trigger** — This trigger bit controls whether the FlexRay module writes continuously or only one pair of Sync Frame Tables into the FRM. If this trigger is set to 1 while SDVEN or SIDEN is set to 1, the FlexRay module writes only one pair of the enabled Sync Frame Tables corresponding to the next even-odd-cycle pair into the FRM. In this case, the FlexRay module clears the SDVEN or SIDEN bits immediately. If this trigger is set to 0 while SDVEN or SIDEN is set to 1, the FlexRay module writes continuously the enabled Sync Frame Tables into the FRM. 0 Write continuously pairs of enabled Sync Frame Tables into FlexRay memory. 1 Write only one pair of enabled Sync Frame Tables into FlexRay memory.
1 SDVEN	**Sync Frame Deviation Table Enable** — This bit controls the generation of the Sync Frame Deviation Tables. The application must set this bit to request the FlexRay module to write the Sync Frame Deviation Tables into the FRM. 0 Do not write Sync Frame Deviation Tables 1 Write Sync Frame Deviation Tables into FRM **Note:** If SDVEN is set to 1, then SIDEN must also be set to 1.
0 SIDEN	**Sync Frame ID Table Enable** — This bit controls the generation of the Sync Frame ID Tables. The application must set this bit to 1 to request the FlexRay module to write the Sync Frame ID Tables into the FRM. 0 Do not write Sync Frame ID Tables 1 Write Sync Frame ID Tables into FRM

3.3.2.32 Sync Frame ID Rejection Filter Register (SFIDRFR)

0x0046 — 16-bit write access required — Write: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	SYNFRID									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-31. Sync Frame ID Rejection Filter Register (SFIDRFR)

This register defines the Sync Frame Rejection Filter ID. The application must update this register outside of the static segment. If the application updates this register in the static segment, it can appear that the FlexRay module accepts the sync frame in the current cycle.

Table 3-39. SFIDRFR Field Descriptions

Field	Description
9–0 SYNFRID	**Sync Frame Rejection ID** — This field defines the frame ID of a frame that must not be used for clock synchronization. For details see Section 3.4.15.2, “Sync Frame Rejection Filtering”.

3.3.2.33 Sync Frame ID Acceptance Filter Value Register (SFIDAFVR)

0x0048 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	FVAL									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-32. Sync Frame ID Acceptance Filter Value Register (SFIDAFVR)

This register defines the sync frame acceptance filter value. For details on filtering, see Section 3.4.15, “Sync Frame Filtering”.

Table 3-40. SFIDAFVR Field Descriptions

Field	Description
9–0 FVAL	**Filter Value** — This field defines the value for the sync frame acceptance filtering.

3.3.2.34 Sync Frame ID Acceptance Filter Mask Register (SFIDAFMR)

0x004A Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	FMSK									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-33. Sync Frame ID Acceptance Filter Mask Register (SFIDAFMR)

This register defines the sync frame acceptance filter mask. For details on filtering see Section 3.4.15.1, “Sync Frame Acceptance Filtering”.

Table 3-41. SFIDAFMR Field Descriptions

Field	Description
9–0 FMSK	**Filter Mask** — This field defines the mask for the sync frame acceptance filtering.

3.3.2.35 Network Management Vector Registers (NMVR0–NMVR5)

0x004C (NMVR0)
0x004E (NMVR1)
0x0050 (NMVR2)
0x0052 (NMVR3)
0x0054 (NMVR4)
0x0056 (NMVR5)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NMVP[15:8]								NMVP[7:0]							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-34. Network Management Vector Registers (NMVR0–NMVR5)

Each of these six registers holds one part of the Network Management Vector. The length of the Network Management Vector is configured in the Network Management Vector Length Register (NMVLR). If NMVLR is programmed with a value less than 12 bytes, the remaining bytes of the Network Management Vector Registers (NMVR0–NMVR5), which are not used for the Network Management Vector accumulating, remains 0s.

The NMVR provides accrued information over all received NMVs in the last communication cycle. All NMVs received in one cycle are ORed into the NMVR. The NMVR is updated at the end of the communication cycle.

Table 3-42. NMVR[0:5] Field Descriptions

Field	Description
15–0 NMVP	**Network Management Vector Part** — The mapping between the Network Management Vector Registers (NMVR0–NMVR5) and the receive message buffer payload bytes in NMV[0:11] is depicted in **Table 3-43**.

Table 3-43. Mapping of NMVRn to the Received Payload Bytes NMVn

NMVRn Register	NMVn Received Payload
NMVR0.NMVP[15:8]	NMV0
NMVR0.NMVP[7:0]	NMV1
NMVR1.NMVP[15:8]	NMV2
NMVR1.NMVP[7:0]	NMV3
...	
NMVR5.NMVP[15:8]	NMV10
NMVR5.NMVP[7:0]	NMV11

3.3.2.36 Network Management Vector Length Register (NMVLR)

0x0058 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	NMVL			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-35. Network Management Vector Length Register (NMVLR)

This register defines the length of the network management vector in bytes.

Table 3-44. NMVLR Field Descriptions

Field	Description
3–0 NMVL	**Network Management Vector Length** — protocol related variable: *gNetworkManagementVectorLength* This field defines the length of the Network Management Vector in bytes. Legal values are between 0 and 12.

3.3.2.37 Timer Configuration and Control Register (TICCR)

0x005A

Write: T2_CFG: *POC:config*
T2_REP, T1_REP, T1SP, T2SP, T1TR, T2TR: Normal Mode

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	T2_CFG	T2_REP	0	0	0	T2ST	0	0	0	T1_REP	0	0	0	T1ST
W						T2SP	T2TR							T1SP	T1TR	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-36. Timer Configuration and Control Register (TICCR)

This register is used to configure and control the two timers, T1 and T2. For timer details, see Section 3.4.17, “Timer Support”. Timer T1 is an absolute timer. Timer T2 can be configured as an absolute or relative timer.

Table 3-45. TICCR Field Descriptions

Field	Description
13 T2_CFG	**Timer T2 Configuration** — This bit configures the timebase mode of Timer T2. 0 T2 is absolute timer. 1 T2 is relative timer.
12 T2_REP	**Timer T2 Repetitive Mode** — This bit configures the repetition mode of Timer T2. 0 T2 is non repetitive 1 T2 is repetitive
10 T2SP	**Timer T2 Stop** — This trigger bit is used to stop timer T2. 0 no effect 1 stop timer T2
9 T2TR	**Timer T2 Trigger** — This trigger bit is used to start timer T2. 0 no effect 1 start timer T2
8 T2ST	**Timer T2 State** — This status bit provides the current state of timer T2. 0 timer T2 is idle 1 timer T2 is running
4 T1_REP	**Timer T1 Repetitive Mode** — This bit configures the repetition mode of timer T1. 0 T1 is non repetitive 1 T1 is repetitive
2 T1SP	**Timer T1 Stop** — This trigger bit is used to stop timer T1. 0 no effect 1 stop timer T1
1 T1TR	**Timer T1 Trigger** — This trigger bit is used to start timer T1. 0 no effect 1 start timer T1
0 T1ST	**Timer T1 State** — This status bit provides the current state of timer T1. 0 timer T1 is idle 1 timer T1 is running

NOTE

Both timers are deactivated immediately when the protocol enters a state different from *POC:normal active* or *POC:normal passive*.

3.3.2.38 Timer 1 Cycle Set Register (TI1CYSR)

0x005C — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	T1_CYC_VAL						0	0	T1_CYC_MSK					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-37. Timer 1 Cycle Set Register (TI1CYSR)

This register defines the cycle filter value and the cycle filter mask for timer T1. For a detailed description of timer T1, refer to Section 3.4.17.1, “Absolute Timer T1”.

Table 3-46. TI1CYSR Field Descriptions

Field	Description
13–8 T1_CYC_VAL	**Timer T1 Cycle Filter Value** — This field defines the cycle filter value for timer T1.
5–0 T1_CYC_MSK	**Timer T1 Cycle Filter Mask** — This field defines the cycle filter mask for timer T1.

NOTE

If the application modifies the value in this register while the timer is running, the change becomes effective immediately and timer T1 expires according to the changed value.

3.3.2.39 Timer 1 Macrotick Offset Register (TI1MTOR)

0x005E — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	T1_MTOFFSET													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-38. Timer 1 Macrotick Offset Register (TI1MTOR)

This register holds the macrotick offset value for timer T1. For a detailed description of timer T1, refer to Section 3.4.17.1, “Absolute Timer T1”.

Table 3-47. TI1MTOR Field Descriptions

Field	Description
13–0 T1_MTOFFSET	**Timer 1 Macrotick Offset** — This field defines the macrotick offset value for timer 1.

NOTE

If the application modifies the value in this register while the timer is running, the change becomes effective immediately and timer T1 expires according to the changed value.

3.3.2.40 Timer 2 Configuration Register 0 (TI2CR0)

0x0060 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	T2_CYC_VAL						0	0	T2_CYC_MSK					
W																
R	T2_MTCNT[31:16]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-39. Timer 2 Configuration Register 0 (TI2CR0)

The content of this register depends on the value of the T2_CFG bit in the Timer Configuration and Control Register (TICCR). For a detailed description of timer T2, refer to Section 3.4.17.2, “Absolute / Relative Timer T2”.

Table 3-48. TI2CR0 Field Descriptions

Field	Description
Fields for absolute timer T2 (TICCR[T2_CFG] = 0)	
13–8 T2_CYC_VAL	**Timer T2 Cycle Filter Value** — This field defines the cycle filter value for timer T2.
5–0 T2_CYC_MSK	**Timer T2 Cycle Filter Mask** — This field defines the cycle filter mask for timer T2.
Fields for relative timer T2 (TICCR[T2_CFG] = 1)	
15–0 T2_MTCNT[31:16]	**Timer T2 Macrotick High Word** — This field defines the high word of the macrotick count for timer T2.

NOTE

If timer T2 is configured as an *absolute* timer and the application modifies the values in this register while the timer is running, the change becomes effective immediately and timer T2 expires according to the changed values.

If timer T2 is configured as a *relative* timer and the application changes the values in this register while the timer is running, the change becomes effective when the timer has expired according to the old values.

3.3.2.41 Timer 2 Configuration Register 1 (TI2CR1)

0x0062 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	T2_MTOFFSET													
W																
R	T2_MTCNT[15:0]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-40. Timer 2 Configuration Register 1 (TI2CR1)

The content of this register depends on the value of the T2_CFG bit in the Timer Configuration and Control Register (TICCR). For a detailed description of timer T2, refer to Section 3.4.17.2, “Absolute / Relative Timer T2”.

Table 3-49. TI2CR1 Field Descriptions

Field	Description
Fields for absolute timer T2 (TICCR[T2_CFG] = 0)	
13–0 T2_MTOFFSET	**Timer T2 Macrotick Offset** — This field holds the macrotick offset value for timer T2.
Fields for relative timer T2 (TICCR[T2_CFG] = 1)	
15–0 T2_MTCNT[15:0]	**Timer T2 Macrotick Low Word** — This field defines the low word of the macrotick value for timer T2.

NOTE

If timer T2 is configured as an *absolute* timer and the application modifies the values in this register while the timer is running, the change becomes effective immediately and the timer T2 expires according to the changed values.

If timer T2 is configured as a *relative* timer and the application changes the values in this register while the timer is running, the change becomes effective when the timer has expired according to the old values.

3.3.2.42 Slot Status Selection Register (SSSR)

0x0064 — 16-bit write access required — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SEL		0	SLOTNUMBER										
W	WMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-41. Slot Status Selection Register (SSSR)

This register is used to access the four internal non memory-mapped slot status selection registers SSSR0 to SSSR3. Each internal registers selects a slot, or symbol window/NIT, whose status vector is saved in the corresponding Slot Status Registers (SSR0–SSR7) according to Table 3-51. For a detailed description of slot status monitoring, refer to Section 3.4.18, “Slot Status Monitoring”.

Table 3-50. SSSR Field Descriptions

Field	Description
15 WMD	**Write Mode** — This control bit defines the write mode of this register. 0 Write to all fields in this register on write access. 1 Write to SEL field only on write access.
13–12 SEL	**Selector** — This field selects one of the four internal slot status selection registers for access. 00 select SSSR0. 01 select SSSR1. 10 select SSSR2. 11 select SSSR3.
10–0 SLOTNUMBER	**Slot Number** — This field specifies the number of the slot whose status is saved in the corresponding slot status registers. **Note:** If this value is set to 0, the related slot status register provides the status of the symbol window after the NIT start, and provides the status of the NIT after the cycle start.

Table 3-51. Mapping Between SSSRn and SSRn

Internal Slot Status Selection Register	Write the Slot Status of the Slot Selected by SSSRn for each			
	Even Communication Cycle		Odd Communication Cycle	
	For Channel B to	For Channel A to	For Channel B to	For Channel A to
SSSR0	SSR0[15:8]	SSR0[7:0]	SSR1[15:8]	SSR1[7:0]
SSSR1	SSR2[15:8]	SSR2[7:0]	SSR3[15:8]	SSR3[7:0]
SSSR2	SSR4[15:8]	SSR4[7:0]	SSR5[15:8]	SSR5[7:0]
SSSR3	SSR6[15:8]	SSR6[7:0]	SSR7[15:8]	SSR7[7:0]

3.3.2.43 Slot Status Counter Condition Register (SSCCR)

0x0066 — 16-bit write access required — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SEL		0	CNTCFG		MCY	VFR	SYF	NUF	SUF	STATUSMASK			
W	WMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-42. Slot Status Counter Condition Register (SSCCR)

This register is used to access and program the four internal non-memory mapped Slot Status Counter Condition Registers SSCCR0 to SSCCR3. Each of these four internal slot status counter condition registers defines the mode and the conditions for incrementing the counter in the corresponding Slot Status Counter Registers (SSCR0–SSCR3). The correspondence is given in Table 3-53. For a detailed description of slot status counters, refer to Section 3.4.18.4, “Slot Status Counter Registers”.

Table 3-52. SSCCR Field Descriptions

Field	Description
15 WMD	**Write Mode** — This control bit defines the write mode of this register. 0 Write to all fields in this register on write access. 1 Write to SEL field only on write access.
13–12 SEL	**Selector** — This field selects one of the four internal slot counter condition registers for access. 00 select SSCCR0. 01 select SSCCR1. 10 select SSCCR2. 11 select SSCCR3.
10–9 CNTCFG	**Counter Configuration** — These bit field controls the channel related incrementing of the slot status counter. 00 increment by 1 if condition is fulfilled on channel A. 01 increment by 1 if condition is fulfilled on channel B. 10 increment by 1 if condition is fulfilled on at least one channel. 11 increment by 2 if condition is fulfilled on both channels channel. increment by 1 if condition is fulfilled on only one channel.
8 MCY	**Multi Cycle Selection** — This bit defines whether the slot status counter accumulates over multiple communication cycles or provides information for the previous communication cycle only. 0 The Slot Status Counter provides information for the previous communication cycle only. 1 The Slot Status Counter accumulates over multiple communication cycles.
7 VFR	**Valid Frame Restriction** — This bit is used to restrict the counter to received valid frames. 0 The counter is not restricted to valid frames only. 1 The counter is restricted to valid frames only.
6 SYF	**Sync Frame Restriction** — This bit is used to restrict the counter to received frames with the sync frame indicator bit set to 1. 0 The counter is not restricted with respect to the sync frame indicator bit. 1 The counter is restricted to frames with the sync frame indicator bit set to 1.
5 NUF	**Null Frame Restriction** — This bit is used to restrict the counter to received frames with the null frame indicator bit set to 0. 0 The counter is not restricted with respect to the null frame indicator bit. 1 The counter is restricted to frames with the null frame indicator bit set to 0.
4 SUF	**Startup Frame Restriction** — This bit is used to restrict the counter to received frames with the startup frame indicator bit set to 1. 0 The counter is not restricted with respect to the startup frame indicator bit. 1 The counter is restricted to received frames with the startup frame indicator bit set to 1.
3–0 STATUSMASK	**Slot Status Mask** — This bit field is used to enable the counter with respect to the four slot status error indicator bits. **STATUSMASK[3]** – This bit enables the counting for slots with the syntax error indicator bit set to 1. **STATUSMASK[2]** – This bit enables the counting for slots with the content error indicator bit set to 1. **STATUSMASK[1]** – This bit enables the counting for slots with the boundary violation indicator bit set to 1. **STATUSMASK[0]** – This bit enables the counting for slots with the transmission conflict indicator bit set to 1.

Table 3-53. Mapping between internal SSCCRn and SSCRn

Condition Register	Condition Defined for Register
SSCCR0	SSCR0
SSCCR1	SSCR1
SSCCR2	SSCR2
SSCCR3	SSCR3

3.3.2.44 Slot Status Registers (SSR0–SSR7)

0x0068 (SSR0)
0x006A (SSR1)
0x006C (SSR2)
0x006E (SSR3)
0x0070 (SSR4)
0x0072 (SSR5)
0x0074 (SSR6)
0x0076 (SSR7)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFB	SYB	NFB	SUB	SEB	CEB	BVB	TCB	VFA	SYA	NFA	SUA	SEA	CEA	BVA	TCA
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-43. Slot Status Registers (SSR0–SSR7)

Each of these eight registers holds the status vector of the slot specified in the corresponding internal slot status selection register, which can be programmed using the Slot Status Selection Register (SSSR). Each register is updated after the end of the corresponding slot as shown in Figure 3-138. The register bits are directly related to the protocol variables and described in more detail in Section 3.4.18, “Slot Status Monitoring”.

Table 3-54. SSR0–SSR7 Field Descriptions

Field	Description
15 VFB	**Valid Frame on Channel B** — protocol related variable: *vSS!ValidFrame* channel B 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
14 SYB	**Sync Frame Indicator Channel B** — protocol related variable: *vRF!Header!SyFIndicator* channel B 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
13 NFB	**Null Frame Indicator Channel B** — protocol related variable: *vRF!Header!NFIndicator* channel B 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
12 SUB	**Startup Frame Indicator Channel B** — protocol related variable: *vRF!Header!SuFIndicator* channel B 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1
11 SEB	**Syntax Error on Channel B** — protocol related variable: *vSS!SyntaxError* channel B 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
10 CEB	**Content Error on Channel B** — protocol related variable: *vSS!ContentError* channel B 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1
9 BVB	**Boundary Violation on Channel B** — protocol related variable: *vSS!BViolation* channel B 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1
8 TCB	**Transmission Conflict on Channel B** — protocol related variable: *vSS!TxConflict* channel B 0 *vSS!TxConflict* = 0 1 *vSS!TxConflict* = 1

Table 3-54. SSR0–SSR7 Field Descriptions (Continued)

Field	Description
7 VFA	**Valid Frame on Channel A** — protocol related variable: *vSS!ValidFrame* channel A 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
6 SYA	**Sync Frame Indicator Channel A** — protocol related variable: *vRF!Header!SyFIndicator* channel A 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
5 NFA	**Null Frame Indicator Channel A** — protocol related variable: *vRF!Header!NFIndicator* channel A 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
4 SUA	**Startup Frame Indicator Channel A** — protocol related variable: *vRF!Header!SuFIndicator* channel A 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1
3 SEA	**Syntax Error on Channel A** — protocol related variable: *vSS!SyntaxError* channel A 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
2 CEA	**Content Error on Channel A** — protocol related variable: *vSS!ContentError* channel A 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1
1 BVA	**Boundary Violation on Channel A** — protocol related variable: *vSS!BViolation* channel A 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1
0 TCA	**Transmission Conflict on Channel A** — protocol related variable: *vSS!TxConflict* channel A 0 *vSS!TxConflict* = 0 1 *vSS!TxConflict* = 1

3.3.2.45 Slot Status Counter Registers (SSCR0–SSCR3)

0x0078 (SSCR0)
0x007A (SSCR1)
0x007C (SSCR2)
0x007E (SSCR3)

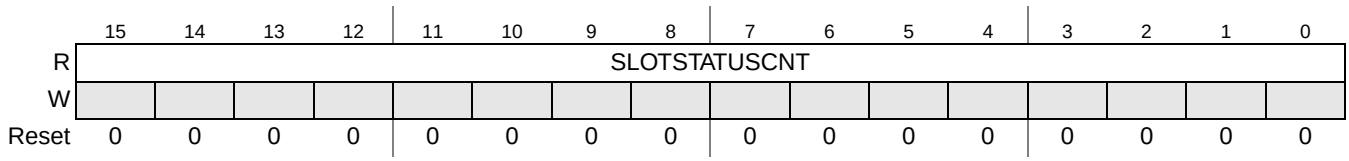


Figure 3-44. Slow Status Counter Registers (SSCR0–SSCR3)

Each of these four registers provides the slot status counter value for the previous communication cycle(s) and is updated at the cycle start. The provided value depends on the control bits and fields in the related internal slot status counter condition register SSCCRn, which can be programmed by using the Slot Status Counter Condition Register (SSCCR). For more details, see Section 3.4.18.4, “Slot Status Counter Registers”.

NOTE

If the counter has reached its maximum value 0xFFFF and is in the multicycle mode (SSCCRn[MCY] equaling 1), the counter is not reset to 0x0000. The application can reset the counter by clearing the SSCCRn[MCY] bit and waiting for the next cycle start, when the FlexRay module clears the counter. Subsequently, the counter can be set into the multicycle mode again.

Table 3-55. SSCR0–SSCR3 Field Descriptions

Field	Description
15–0 SLOTSTATUSCNT	**Slot Status Counter** — This field provides the current value of the Slot Status Counter.

3.3.2.46 MTS A Configuration Register (MTSACFR)

0x0080

Write: MTE: Any Time
CYCCNTMSK, CYCCNTVAL: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MTE	0	CYCCNTMSK						0	0	CYCCNTVAL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-45. MTS A Configuration Register (MTSACFR)

This register controls the transmission of the Media Access Test Symbol MTS on channel A. For more details, see Section 3.4.13, “MTS Generation”.

Table 3-56. MTSACFR Field Descriptions

Field	Description
15 MTE	**Media Access Test Symbol Transmission Enable** — This control bit is used to enable and disable the transmission of the Media Access Test Symbol in the selected set of cycles. 0 MTS transmission disabled 1 MTS transmission enabled
13–8 CYCCNTMSK	**Cycle Counter Mask** — This field provides the filter mask for the MTS cycle count filter.
5–0 CYCCNTVAL	**Cycle Counter Value** — This field provides the filter value for the MTS cycle count filter.

3.3.2.47 MTS B Configuration Register (MTSBCFR)

0x0082

Write: MTE: Any Time
CYCCNTMSK, CYCCNTVAL: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MTE	0	CYCCNTMSK						0	0	CYCCNTVAL					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-46. MTS B Configuration Register (MTSBCFR)

This register controls the transmission of the Media Access Test Symbol MTS on channel B. For more details, see Section 3.4.13, “MTS Generation”.

Table 3-57. MTSBCFR Field Descriptions

Field	Description
15 MTE	**Media Access Test Symbol Transmission Enable** — This control bit is used to enable and disable the transmission of the Media Access Test Symbol in the selected set of cycles. 0 MTS transmission disabled 1 MTS transmission enabled
13–8 CYCCNTMSK	**Cycle Counter Mask** — This field provides the filter mask for the MTS cycle count filter.
5–0 CYCCNTVAL	**Cycle Counter Value** — This field provides the filter value for the MTS cycle count filter.

3.3.2.48 Receive Shadow Buffer Index Register (RSBIR)

0x0084 — 16-bit write access required — Write: WMD, SEL: Any Time; RSBIDX: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	SEL		0	0	0	0	RSBIDX							
W	WMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-47. Receive Shadow Buffer Index Register (RSBIR)

This register is used to provide and retrieve the indices of the message buffer header fields currently associated with the receive shadow buffers. For more details on the receive shadow buffer concept, refer to Section 3.4.6.3.6, “Receive Shadow Buffers Concept”.

Table 3-58. RSBIR Field Descriptions

Field	Description
15 WMD	**Write Mode** — This bit controls the write mode for this register. 0 update SEL and RSBIDX field on register write 1 update only SEL field on register write
13–12 SEL	**Selector** — This field is used to select the internal receive shadow buffer index register for access. 00 RSBIR_A1 — receive shadow buffer index register for channel A, segment 1 01 RSBIR_A2 — receive shadow buffer index register for channel A, segment 2 10 RSBIR_B1 — receive shadow buffer index register for channel B, segment 1 11 RSBIR_B2 — receive shadow buffer index register for channel B, segment 2
7–0 RSBIDX	**Receive Shadow Buffer Index** — This field contains the current index of the message buffer header field of the receive shadow message buffer selected by the SEL field. The FlexRay module uses this index to determine the physical location of the shadow buffer header field in the FlexRay memory. The FlexRay module updates this field during receive operation.The application provides initial message buffer header index value in the configuration phase. FlexRay module: Updates the message buffer header index after successful reception. Application: Provides initial message buffer header index.

3.3.2.49 Receive FIFO Selection Register (RFSR)

0x0086 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	SEL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-48. Receive FIFO Selection Register (RFSR)

This register is used to select a receiver FIFO for subsequent access through the receiver FIFO configuration registers summarized in Table 3-59.

Table 3-59. SEL Controlled Receiver FIFO Registers

Register
Receive FIFO Start Index Register (RFSIR)
Receive FIFO Depth and Size Register (RFDSR)
Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR)
Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR)
Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR)
Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR)
Receive FIFO Range Filter Configuration Register (RFRFCFR)
Receive FIFO Range Filter Control Register (RFRFCTR)

Table 3-60. RFSR Field Descriptions

Field	Description
0 SEL	**Select** — This control bit selects the receiver FIFO for subsequent programming. 0 Receiver FIFO for channel A selected 1 Receiver FIFO for channel B selected

3.3.2.50 Receive FIFO Start Index Register (RFSIR)

0x0088 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	SIDX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-49. Receive FIFO Start Index Register (RFSIR)

This register defines the message buffer header index of the first message buffer of the selected FIFO.

Table 3-61. RFSIR Field Descriptions

Field	Description
7–0 SIDX	**Start Index** — This field defines the number of the message buffer header field of the first message buffer of the selected receive FIFO. The FlexRay module uses the value of the SIDX field to determine the physical location of the receiver FIFO's first message buffer header field.

3.3.2.51 Receive FIFO Depth and Size Register (RFDSR)

0x008A — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIFO_DEPTH								0	ENTRY_SIZE						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-50. Receive FIFO Depth and Size Register (RFDSR)

This register defines the structure of the selected FIFO, i.e. the number of entries and the size of each entry.

Table 3-62. RFDSR Field Descriptions

Field	Description
15–8 FIFO_DEPTH	**FIFO Depth** — This field defines the depth of the selected receive FIFO, i.e. the number of entries.
6–0 ENTRY_SIZE	**Entry Size** — This field defines the size of the frame data sections for the selected receive FIFO in 2 byte entities.

3.3.2.52 Receive FIFO A Read Index Register (RFARIR)

0x008C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	RDIDX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-51. Receive FIFO A Read Index Register (RFARIR)

This register provides the message buffer header index of the next available receive FIFO A entry that the application can read.

Table 3-63. RFARIR Field Descriptions

Field	Description
7–0 RDIDX	**Read Index** — This field provides the message buffer header index of the next available receive FIFO message buffer that the application can read. The FlexRay module increments this index when the application writes to the FNEAIF flag in the Global Interrupt Flag and Enable Register (GIFER). The index wraps back to the first message buffer header index if the end of the FIFO was reached.

NOTE

If the receive FIFO not empty flag FNEAIF is not set, the RDIDX field points to a physical message buffer that contains invalid data. Only when FNEAIF is set, does the message buffer indicated by RDIDX contain valid data.

3.3.2.53 Receive FIFO B Read Index Register (RFBRIR)

0x008E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	RDIDX							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-52. Receive FIFO B Read Index Register (RFBRIR)

This register provides the message buffer header index of the next available receive FIFO B entry that the application can read.

Table 3-64. RFBRIR Field Descriptions

Field	Description
7–0 RDIDX	**Read Index** — This field provides the message buffer header index of the next available receive FIFO entry that the application can read. The FlexRay module increments this index when the application writes to the FNEBIF flag in the Global Interrupt Flag and Enable Register (GIFER).The index wraps back to the first message buffer header index if the end of the FIFO was reached.

NOTE

If the receive FIFO not empty flag FNEBIF is not set, the RDIDX field points to a physical message buffer that contains invalid data. Only when FNEBIF is set, does the message buffer indicated by RDIDX contain valid data.

3.3.2.54 Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR)

0x0090 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MIDAFVAL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-53. Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR)

This register defines the filter value for the message ID acceptance filter of the selected receive FIFO. For details on message ID filtering see Section 3.4.9.5, “Receive FIFO filtering”.

Table 3-65. RFMIDAFVR Field Descriptions

Field	Description
15–0 MIDAFVAL	**Message ID Acceptance Filter Value** — Filter value for the message ID acceptance filter.

3.3.2.55 Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR)

0x0092 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MIDAFMSK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-54. Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR)

This register defines the filter mask for the message ID acceptance filter of the selected receive FIFO. For details on message ID filtering see Section 3.4.9.5, “Receive FIFO filtering”.

Table 3-66. RFMIAFMR Field Descriptions

Field	Description
15–0 MIDAFMSK	**Message ID Acceptance Filter Mask** — Filter mask for the message ID acceptance filter.

3.3.2.56 Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR)

0x0094 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	FIDRFVAL										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-55. Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR)

This register defines the filter value for the frame ID rejection filter of the selected receive FIFO. For details on frame ID filtering see Section 3.4.9.5, “Receive FIFO filtering”.

Table 3-67. RFFIDRFVR Field Descriptions

Field	Description
10–0 FIDRFVAL	**Frame ID Rejection Filter Value** — Filter value for the frame ID rejection filter.

3.3.2.57 Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR)

0x0096 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	FIDRFMSK										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-56. Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR)

This register defines the filter mask for the frame ID rejection filter of the selected receive FIFO. For details on frame ID filtering see Section 3.4.9.5, “Receive FIFO filtering”.

Table 3-68. RFFIDRFMR Field Descriptions

Field	Description
10–0 FIDRFMSK	**Frame ID Rejection Filter Mask** — Filter mask for the frame ID rejection filter.

3.3.2.58 Receive FIFO Range Filter Configuration Register (RFRFCFR)

0x0098 16-bit write access required Write: WMD, IBD, SEL: Any Time
SID: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	IBD	SEL		0	SID										
W	WMD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-57. Receive FIFO Range Filter Configuration Register (RFRFCFR)

This register provides access to the four internal frame ID range filter boundary registers of the selected receive FIFO. For details on frame ID range filter see Section 3.4.9.5, “Receive FIFO filtering”.

Table 3-69. RFRFCFR Field Descriptions

Field	Description
15 WMD	**Write Mode** — This control bit defines the write mode of this register. 0 Write to all fields in this register on write access. 1 Write to SEL and IBD field only on write access.
14 IBD	**Interval Boundary** — This control bit selects the interval boundary to be programmed with the SID value. 0 program lower interval boundary 1 program upper interval boundary
13–12 SEL	**Filter Selector** — This control field selects the frame ID range filter to be accessed. 00 select frame ID range filter 0. 01 select frame ID range filter 1. 10 select frame ID range filter 2. 11 select frame ID range filter 3.
10–0 SID	**Slot ID —** Defines the IBD-selected frame ID boundary value for the SEL-selected range filter.

3.3.2.59 Receive FIFO Range Filter Control Register (RFRFCTR)

0x009A Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	F3MD	F2MD	F1MD	F0MD	0	0	0	0	F3EN	F2EN	F1EN	F0EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-58. Receive FIFO Range Filter Control Register (RFRFCTR)

This register is used to enable and disable each frame ID range filter and to define whether it is running as acceptance or rejection filter.

Table 3-70. RFRFCTR Field Descriptions

Field	Description
11 F3MD	**Range Filter 3 Mode** — This control bit defines the filter mode of the frame ID range filter 3. 0 range filter 3 runs as acceptance filter 1 range filter 3 runs as rejection filter
10 F2MD	**Range Filter 2 Mode** — This control bit defines the filter mode of the frame ID range filter 2. 0 range filter 2 runs as acceptance filter 1 range filter 2 runs as rejection filter
9 F1MD	**Range Filter 1 Mode** — This control bit defines the filter mode of the frame ID range filter 1. 0 range filter 1 runs as acceptance filter 1 range filter 1 runs as rejection filter
8 F0MD	**Range Filter 0 Mode** — This control bit defines the filter mode of the frame ID range filter 0. 0 range filter 0 runs as acceptance filter 1 range filter 0 runs as rejection filter
3 F3EN	**Range Filter 3 Enable** — This control bit is used to enable and disable the frame ID range filter 3. 0 range filter 3 disabled 1 range filter 3 enabled
2 F2EN	**Range Filter 2 Enable** — This control bit is used to enable and disable the frame ID range filter 2. 0 range filter 2 disabled 1 range filter 2 enabled
1 F1EN	**Range Filter 1 Enable** — This control bit is used to enable and disable the frame ID range filter 1. 0 range filter 1 disabled 1 range filter 1 enabled
0 F0EN	**Range Filter 0 Enable** — This control bit is used to enable and disable the frame ID range filter 0. 0 range filter 0 disabled 1 range filter 0 enabled

3.3.2.60 Last Dynamic Transmit Slot Channel A Register (LDTXSLAR)

0x009C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LASTDYNTXSLOTA										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-59. Last Dynamic Slot Channel A Register (LDTXSLAR)

This register provides the number of the last transmission slot in the dynamic segment for channel A. This register is updated after the end of the dynamic segment and before the start of the next communication cycle.

Table 3-71. LDTXSLAR Field Descriptions

Field	Description
10–0 LASTDYNTX SLOTA	**Last Dynamic Transmission Slot Channel A** — protocol related variable: *zLastDynTxSlot* channel A Number of the last transmission slot in the dynamic segment for channel A. If no frame was transmitted during the dynamic segment on channel A, the value of this field is set to 0.

3.3.2.61 Last Dynamic Transmit Slot Channel B Register (LDTXSLBR)

0x009E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	LASTDYNTXSLOTB										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-60. Last Dynamic Slot Channel B Register (LDTXSLBR)

This register provides the number of the last transmission slot in the dynamic segment for channel B. This register is updated after the end of the dynamic segment and before the start of the next communication cycle.

Table 3-72. LDTXSLBR Field Descriptions

Field	Description
10–0 LASTDYNTXSLOTB	**Last Dynamic Transmission Slot Channel B** — protocol related variable: *zLastDynTxSlot* channel B Number of the last transmission slot in the dynamic segment for channel B. If no frame was transmitted during the dynamic segment on channel B the value of this field is set to 0.

3.3.2.62 Protocol Configuration Registers

The following configuration registers provide the necessary configuration information to the protocol engine. The individual values in the registers are described in Table 3-73. For more details about the FlexRay related configuration parameters and the allowed parameter ranges, see *FlexRay Communications System Protocol Specification, Version 2.1 Rev A.*

Table 3-73. Protocol Configuration Register Fields

Name	Description[1]	Min	Max	Unit	PCR
coldstart_attempts	*gColdstartAttempts*			number	3
action_point_offset	*gdActionPointOffset* - 1			MT	0
cas_rx_low_max	*gdCASRxLowMax* - 1			*gdBit*	4
dynamic_slot_idle_phase	*gdDynamicSlotIdlePhase*			minislot	28
minislot_action_point_offset	*gdMinislotActionPointOffset* - 1			MT	3
minislot_after_action_point	*gdMinislot* - *gdMinislotActionPointOffset* - 1			MT	2
static_slot_length	*gdStaticSlot*			MT	0
static_slot_after_action_point	*gdStaticSlot* - *gdActionPointOffset* - 1			MT	13
symbol_window_exists	*gdSymbolWindow*!=0	0	1	bool	9
symbol_window_after_action_point	*gdSymbolWindow* - *gdActionPointOffset* - 1			MT	6
tss_transmitter	*gdTSSTransmitter*			*gdBit*	5
wakeup_symbol_rx_idle	*gdWakeupSymbolRxIdle*			*gdBit*	5
wakeup_symbol_rx_low	*gdWakeupSymbolRxLow*			*gdBit*	3
wakeup_symbol_rx_window	*gdWakeupSymbolRxWindow*			*gdBit*	4
wakeup_symbol_tx_idle	*gdWakeupSymbolTxIdle*			*gdBit*	8
wakeup_symbol_tx_low	*gdWakeupSymbolTxLow*			*gdBit*	5
noise_listen_timeout	(*gListenNoise* * *pdListenTimeout*) - 1			µT	16/17

Table 3-73. Protocol Configuration Register Fields (Continued)

Name	Description[1]	Min	Max	Unit	PCR
macro_initial_offset_a	*pMacroInitialOffset[A]*			MT	6
macro_initial_offset_b	*pMacroInitialOffset[B]*			MT	16
macro_per_cycle	*gMacroPerCycle*			MT	10
macro_after_first_static_slot	*gMacroPerCycle* - *gdStaticSlot*			MT	1
macro_after_offset_correction	*gMacroPerCycle* - *gOffsetCorrectionStart*			MT	28
max_without_clock_correction_fatal	*gMaxWithoutClockCorrectionFatal*			cyclepairs	8
max_without_clock_correction_passive	*gMaxWithoutClockCorrectionPassive*			cyclepairs	8
minislot_exists	*gNumberOfMinislots*!=0	0	1	bool	9
minislots_max	*gNumberOfMinislots* - 1			minislot	29
number_of_static_slots	*gNumberOfStaticSlots*			static slot	2
offset_correction_start	*gOffsetCorrectionStart*			MT	11
payload_length_static	*gPayloadLengthStatic*			2-bytes	19
max_payload_length_dynamic	*pPayloadLengthDynMax*			2-bytes	24
first_minislot_action_point_offset	max(*gdActionPointOffset*, *gdMinislotActionPointOffset*) - 1			MT	13
allow_halt_due_to_clock	*pAllowHaltDueToClock*			bool	26
allow_passive_to_active	*pAllowPassiveToActive*			cyclepairs	12
cluster_drift_damping	*pClusterDriftDamping*			µT	24
comp_accepted_startup_range_a	*pdAcceptedStartupRange* - *pDelayCompensationChA*			µT	22
comp_accepted_startup_range_b	*pdAcceptedStartupRange* - *pDelayCompensationChB*			µT	26
listen_timeout	*pdListenTimeout* - 1			µT	14/15
key_slot_id	*pKeySlotId*			number	18
key_slot_used_for_startup	*pKeySlotUsedForStartup*			bool	11
key_slot_used_for_sync	*pKeySlotUsedForSync*			bool	11
latest_tx	*gNumberOfMinislots* - *pLatestTx*			minislot	21
sync_node_max	*gSyncNodeMax*			number	30
micro_initial_offset_a	*pMicroInitialOffset[A]*			µT	20
micro_initial_offset_b	*pMicroInitialOffset[B]*			µT	20
micro_per_cycle	*pMicroPerCycle*			µT	22/23
micro_per_cycle_min	*pMicroPerCycle - pdMaxDrift*			µT	24/25
micro_per_cycle_max	*pMicroPerCycle + pdMaxDrift*			µT	26/27
micro_per_macro_nom_half	round(*pMicroPerMacroNom* / 2)			µT	7
offset_correction_out	*pOffsetCorrectionOut*			µT	9
rate_correction_out	*pRateCorrectionOut*			µT	14
single_slot_enabled	*pSingleSlotEnabled*			bool	10
wakeup_channel	*pWakeupChannel*	see Table 3-74			10
wakeup_pattern	*pWakeupPattern*			number	18
decoding_correction_a	*pDecodingCorrection* + *pDelayCompensation[A]* + 2			µT	19

Table 3-73. Protocol Configuration Register Fields (Continued)

Name	Description[1]	Min	Max	Unit	PCR
decoding_correction_b	*pDecodingCorrection* + *pDelayCompensation[B]* + 2			µT	7
key_slot_header_crc	header CRC for key slot	0x000	0x7FF	number	12
extern_offset_correction	*pExternOffsetCorrection*			µT	29
extern_rate_correction	*pExternRateCorrection*			µT	21

1 See *FlexRay Communications System Protocol Specification, Version 2.1 Rev A* for detailed protocol parameter definitions

Table 3-74. Wakeup Channel Selection

wakeup_channel	Wakeup Channel
0	A
1	B

3.3.2.62.1 Protocol Configuration Register 0 (PCR0)

0x00A0 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	action_point_offset								static_slot_length							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-61. Protocol Configuration Register 0 (PCR0)

3.3.2.62.2 Protocol Configuration Register 1 (PCR1)

0x00A2 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	macro_after_first_static_slot													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-62. Protocol Configuration Register 1 (PCR1)

3.3.2.62.3 Protocol Configuration Register 2 (PCR2)

0x00A4 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	minislot_after_action_point								number_of_static_slots							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-63. Protocol Configuration Register 2 (PCR2)

3.3.2.62.4 Protocol Configuration Register 3 (PCR3)

0x00A6 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	wakeup_symbol_rx_low								minislot_action_point_offset[4:0]				coldstart_attempts			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-64. Protocol Configuration Register 3 (PCR3)

3.3.2.62.5 Protocol Configuration Register 4 (PCR4)

0x00A8 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cas_rx_low_max								wakeup_symbol_rx_window							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-65. Protocol Configuration Register 4 (PCR4)

3.3.2.62.6 Protocol Configuration Register 5 (PCR5)

0x00AA Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tss_transmitter				wakeup_symbol_tx_low						wakeup_symbol_rx_idle					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-66. Protocol Configuration Register 5 (PCR5)

3.3.2.62.7 Protocol Configuration Register 6 (PCR6)

0x00AC Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	symbol_window_after_action_point								macro_initial_offset_a						
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-67. Protocol Configuration Register 6 (PCR6)

3.3.2.62.8 Protocol Configuration Register 7 (PCR7)

0x00AE Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	decoding_correction_b								micro_per_macro_nom_half							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-68. Protocol Configuration Register 7 (PCR7)

3.3.2.62.9 Protocol Configuration Register 8 (PCR8)

0x00B0 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	max_without_clock_correction_fatal				max_without_clock_correction_passive				wakeup_symbol_tx_idle							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-69. Protocol Configuration Register 8 (PCR8)

3.3.2.62.10 Protocol Configuration Register 9 (PCR9)

0x00B2 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mini slot_ exists	symbol_ window_ exists	offset_correction_out													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-70. Protocol Configuration Register 9 (PCR9)

3.3.2.62.11 Protocol Configuration Register 10 (PCR10)

0x00B4 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	single_slot_enabled	wakeup_channel	macro_per_cycle													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-71. Protocol Configuration Register 10 (PCR10)

3.3.2.62.12 Protocol Configuration Register 11 (PCR11)

0x00B6 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	key_slot_used_for_startup	key_slot_used_for_sync	offset_correction_start													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-72. Protocol Configuration Register 11 (PCR11)

3.3.2.62.13 Protocol Configuration Register 12 (PCR12)

0x00B8 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	allow_passive_to_active								key_slot_header_crc							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-73. Protocol Configuration Register 12 (PCR12)

3.3.2.62.14 Protocol Configuration Register 13 (PCR13)

0x00BA Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	first_minislot_action_point_offset								static_slot_after_action_point							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-74. Protocol Configuration Register 13 (PCR13)

3.3.2.62.15 Protocol Configuration Register 14 (PCR14)

0x00BC Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rate_correction_out												listen_timeout[20:16]			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-75. Protocol Configuration Register 14 (PCR14)

3.3.2.62.16 Protocol Configuration Register 15 (PCR15)

0x00BE Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	listen_timeout[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-76. Protocol Configuration Register 15 (PCR15)

3.3.2.62.17 Protocol Configuration Register 16 (PCR16)

0x00C0 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	macro_initial_offset_b								noise_listen_timeout[24:16]							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-77. Protocol Configuration Register 16 (PCR16)

3.3.2.62.18 Protocol Configuration Register 17 (PCR17)

0x00C2 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	noise_listen_timeout[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-78. Protocol Configuration Register 17 (PCR17)

3.3.2.62.19 Protocol Configuration Register 18 (PCR18)

0x00C4 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	wakeup_pattern						key_slot_id									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-79. Protocol Configuration Register 18 (PCR18)

3.3.2.62.20 Protocol Configuration Register 19 (PCR19)

0x00C6 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	decoding_correction_a								payload_length_static							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-80. Protocol Configuration Register 19 (PCR19)

3.3.2.62.21 Protocol Configuration Register 20 (PCR20)

0x00C8 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	micro_initial_offset_b								micro_initial_offset_a							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-81. Protocol Configuration Register 20 (PCR20)

3.3.2.62.22 Protocol Configuration Register 21 (PCR21)

0x00CA Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	extern_rate_ correction				latest_tx											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-82. Protocol Configuration Register 21 (PCR21)

3.3.2.62.23 Protocol Configuration Register 22 (PCR22)

0x00CC — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R*				comp_accepted_startup_range_a								micro_per_cycle[19:16			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-83. Protocol Configuration Register 22 (PCR22)

3.3.2.62.24 Protocol Configuration Register 23 (PCR23)

0x00CE — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	micro_per_cycle[15:0]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-84. Protocol Configuration Register 23 (PCR23)

3.3.2.62.25 Protocol Configuration Register 24 (PCR24)

0x00D0 — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cluster_drift_damping				max_payload_length_dynamic								micro_per_cycle_min [19:16]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-85. Protocol Configuration Register 24 (PCR24)

3.3.2.62.26 Protocol Configuration Register 25 (PCR25)

0x00D2 — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	micro_per_cycle_min[15:0]															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-86. Protocol Configuration Register 25 (PCR25)

3.3.2.62.27 Protocol Configuration Register 26 (PCR26)

0x00D4 — Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	allow_halt_due_to_clock				comp_accepted_startup_range_b								micro_per_cycle_max [19:16]			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-87. Protocol Configuration Register 26 (PCR26)

3.3.2.62.28 Protocol Configuration Register 27 (PCR27)

0x00D6 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	micro_per_cycle_max[15:0]															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-88. Protocol Configuration Register 27 (PCR27)

3.3.2.62.29 Protocol Configuration Register 28 (PCR28)

0x00D8 Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dynamic_slot _idle_phase				macro_after_offset_correction											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-89. Protocol Configuration Register 28 (PCR28)

3.3.2.62.30 Protocol Configuration Register 29 (PCR29)

0x00DA Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	extern_offset_ correction				minislots_max											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-90. Protocol Configuration Register 29 (PCR29)

3.3.2.62.31 Protocol Configuration Register 30 (PCR30)

0x00DC Write: *POC:config*

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	sync_node_max			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-91. Protocol Configuration Register 30 (PCR30)

3.3.2.63 Message Buffer Configuration, Control, Status Registers (MBCCSRn)

0x0100 (MBCCSR0)
0x0108 (MBCCSR1)
...
0x04F8 (MBCCSR127)

Write: MCM, MBT, MTD: *POC:config* or MB_DIS
CMT: MB_LCK
EDT, LCKT, MBIE, MBIF: Normal Mode

Additional Reset: CMT, DUP, DVAL, MBIF: Message Buffer Disable

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	MCM	MBT	MTD	CMT	0	0	MBIE	0	0	0	DUP	DVAL	EDS	LCKS	MBIF
W						EDT	LCKT									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-92. Message Buffer Configuration, Control, Status Registers (MBCCSRn)

The content of these registers comprises message buffer configuration data, message buffer control data, message buffer status information, and message buffer interrupt flags.

Table 3-75. MBCCSRn Field Descriptions

Field	Description
Message Buffer Configuration	
14 MCM	**Message Buffer Commit Mode** — This bit applies only to double buffered transmit message buffers and defines the commit mode. 0 Streaming commit mode 1 Immediate commit mode
13 MBT	**Message Buffer Type** — This bit applies only to transmit message buffers and defines the buffering type. 0 Single buffered transmit message buffer 1 Double buffered transmit message buffer
12 MTD	**Message Buffer Transfer Direction** — This bit defines the transfer direction of the message buffer. 0 Receive message buffer 1 Transmit message buffer
Message Buffer Control	
11 CMT	**Commit for Transmission** — This bit applies only to transmit message buffers and indicates whether the message buffer contains valid data that are ready for transmission. Both the application and the FlexRay module can modify this bit. • Application: The application sets this bit to indicate that the transmit message buffer contains valid data ready for transmission. The application clears this bit to indicate that the message buffer data are no longer valid for transmission. • FlexRay module: The FlexRay module clears this bit when the message buffer data are no longer valid for transmission. 0 Message buffer does not contain valid data. 1 Message buffer contains valid data.
10 EDT	**Enable/Disable Trigger** — This trigger bit is used to enable and disable a message buffer. The message buffer enable is triggered when the application writes 1 to this bit and the message buffer is disabled, i.e. the EDS status bit is 0. The message buffer disable is triggered when the application writes 1 to this bit and the message buffer is enabled, i.e. the EDS status bit is 1. 0 No effect 1 message buffer enable/disable triggered **Note:** If the application writes 1 to this bit, the write access to all other bits is ignored.

Table 3-75. MBCCSRn Field Descriptions (Continued)

Field	Description
9 LCKT	**Lock/Unlock Trigger** — This trigger bit is used to lock and unlock a message buffer. The message buffer lock is triggered when the application writes 1 to this bit and the message buffer is not locked, i.e. the LCKS status bit is 0. The message buffer unlock is triggered when the application writes 1 to this bit and the message buffer is locked, i.e. the LCKS status bit is 1. 0 No effect 1 Trigger message buffer lock/unlock **Note:** If the application writes 1 to this bit and 0 to the EDT bit, the write access to all other bits is ignored.
8 MBIE	**Message Buffer Interrupt Enable** — This control bit defines whether the message buffer generate an interrupt request when its MBIF flag is set. 0 Interrupt request generation disabled 1 Interrupt request generation enabled
Message Buffer Status	
4 DUP	**Data Updated** — This status bit applies only to receive message buffers. It is always 0 for transmit message buffers. This bit provides information whether the frame header in the message buffer header field and the message buffer data field were updated. See Section 3.4.6.3.4, “Message Buffer Status Update” for a detailed description of the update condtions. 0 Frame Header and Message buffer data field not updated. 1 Frame Header and Message buffer data field updated.
3 DVAL	**Data Valid** — The semantic of this status bit depends on the message buffer type and transfer direction. • *Receive Message Buffer:* Indicates whether the message buffer data field contains valid frame data. See Section 3.4.6.3.4, “Message Buffer Status Update” for a detailed update description of the update conditions. 0 message buffer data field contains no valid frame data 1 message buffer data field contains valid frame data • *Single Transmit Message Buffer*: Indicates whether the message is transferred again due to the state transmission mode of the message buffer. 0 Message transferred for the first time. 1 Message is transferred again. • *Double Transmit Message Buffer*: For the commit side it is always 0. For the transmit side it indicates whether the message is transferred again due to the state transmission mode of the message buffer. 0 Message transferred for the first time. 1 Message is transferred again.
2 EDS	**Enable/Disable Status** — This status bit indicates whether the message buffer is enabled or disabled. 0 Message buffer is disabled. 1 Message buffer is enabled.
1 LCKS	**Lock Status** — This status bit indicates the current lock status of the message buffer. 0 Message buffer is not locked by the application. 1 Message buffer is locked by the application.
0 MBIF	**Message Buffer Interrupt Flag** — The semantic of this flag depends on the message buffer transfer direction. • *Receive Message Buffer:* This flag is set when the slot status in the message buffer header field was updated and this slot was not an empty dynamic slot. See Section 3.4.6.3.4, “Message Buffer Status Update” for a detailed description of the update conditions. 0 slot status not updated 1 slot status updated and slot was not an empty dynamic slot • *Transmit Message Buffer:* This flag is set when the slot status in the message buffer header field was updated. Additionally this flag is set immediately when a transmit message buffer was enabled. 0 slot status not updated 1 slot status updated / message buffer recently enabled Writing a 1 clears this flag. Writing a 0 does not change the flag state.

3.3.2.64 Message Buffer Cycle Counter Filter Registers (MBCCFRn)

0x0102 (MBCCFR0)
0x010A (MBCCFR1)
...
, 0x04FA (MBCCFR127)

Write: *POC:config* or MB_DIS

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MTM	CHA	CHB	CCFE	CCFMSK						CCFVAL					
Reset	bits located in physical memory, not affected by reset, no reset value															

Figure 3-93. Message Buffer Cycle Counter Filter Registers (MBCCFRn)

This register contains message buffer configuration data for the transmission mode, the channel assignment, and for the cycle counter filtering. For detailed information on cycle counter filtering, refer to Section 3.4.7.1.2, “Message Buffer Cycle Counter Filtering”.

Table 3-76. MBCCFRn Field Descriptions

Field	Description
15 MTM	**Message Buffer Transmission Mode** — This control bit applies only to transmit message buffers and defines the transmission mode. 0 Event transmission mode 1 State transmission mode
14–13 CHA CHB	**Channel Assignment** — These control bits define the channel assignment and control the receive and transmit behavior of the message buffer according to **Table 3-77**.
12 CCFE	**Cycle Counter Filtering Enable** — This control bit is used to enable and disable the cycle counter filtering. 0 Cycle counter filtering disabled 1 Cycle counter filtering enabled
11–6 CCFMSK	**Cycle Counter Filtering Mask** — This field defines the filter mask for the cycle counter filtering.
5–0 CCFVAL	**Cycle Counter Filtering Value** — This field defines the filter value for the cycle counter filtering.

Table 3-77. Channel Assignment Description

CHA	CHB	Transmit Message Buffer		Receive Message Buffer	
		static segment	dynamic segment	static segment	dynamic segment
1	1	transmit on both channel A and channel B	transmit on channel A only	store first valid frame received on channel A or channel B	store first valid frame received on channel A, ignore channel B
0	1	transmit on channel B	transmit on channel B	store first valid frame received on channel B	store first valid frame received on channel B
1	0	transmit on channel A	transmit on channel A	store first valid frame received on channel A	store first valid frame received on channel A
0	0	no frame transmission	no frame transmission	no frame stored	no frame stored

NOTE

If at least one message buffer assigned to a certain slot is assigned to both channels, then all message buffers assigned to this slot have to be assigned to both channels. Otherwise, the message buffer configuration is illegal and the result of the message buffer search is not defined.

3.3.2.65 Message Buffer Frame ID Registers (MBFIDRn)

0x0104 (MBFIDR0)
0x010C (MBFIDR1)
...
0x04FC (MBFIDR127)

Write: *POC:config* or MB_DIS

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	FID										
W																
Reset	0	0	0	0	0	bits located in physical memory, not affected by reset, no reset value										

Figure 3-94. Message Buffer Frame ID Registers (MBFIDRn)

Table 3-78. MBFIDRn Field Descriptions

Field	Description
10–0 FID	**Frame ID** — The semantic of this field depends on the message buffer transfer type. • *Receive Message Buffer:* This field is used as a filter value to determine if the message buffer is used for reception of a message received in a slot with the slot ID equal to FID. • *Transmit Message Buffer:* This field is used to determine the slot in which the message in this message buffer should be transmitted.

3.3.2.66 Message Buffer Index Registers (MBIDXRn)

0x0106 (MBIDXR0)
0x010E (MBIDXR1)
...
0x04FE (MBIDXR127)

Write: *POC:config* or MB_DIS

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	MBIDX							
W																
Reset	0	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-

Figure 3-95. Message Buffer Index Registers (MBIDXRn)

Table 3-79. MBIDXRn Field Descriptions

Field	Description
7–0 MBIDX	**Message Buffer Index** — This field provides the index of the message buffer header field of the physical message buffer that is currently associated with this message buffer. The application writes the index of the initially associated message buffer header field into this register. The FlexRay module updates this register after frame reception or transmission.

3.4 Functional Description

This section provides a detailed description of the functionality implemented in the FlexRay module.

3.4.1 Message Buffer Concept

The FlexRay module uses a data structure called *message buffer* to store frame data, configuration, control, and status data. Each message buffer consists of two parts, the *message buffer control data* and the *physical message buffer.* The message buffer control data are located in dedicated registers. The structure of the message buffer control data depends on the message buffer type and is described in Section 3.4.3, “Message Buffer Types”. The physical message buffer is located in the FRM and is described in Section 3.4.2, “Physical Message Buffer”.

3.4.2 Physical Message Buffer

All FlexRay messages and related frame and slot status information of received frames and of frames to be transmitted to the FlexRay bus are stored in data structures called *physical message buffers*. The physical message buffers are located in the FRM.The structure of a physical message buffer is depicted in Figure 3-96.

A physical message buffer consists of two fields, the *message buffer header field* and the *message buffer data field*. The message buffer header field contains the *frame header*, the *data field offset*, and the *slot status*.The message buffer data field contains the *frame data*.

The connection between the two fields is established by the *data field offset*.

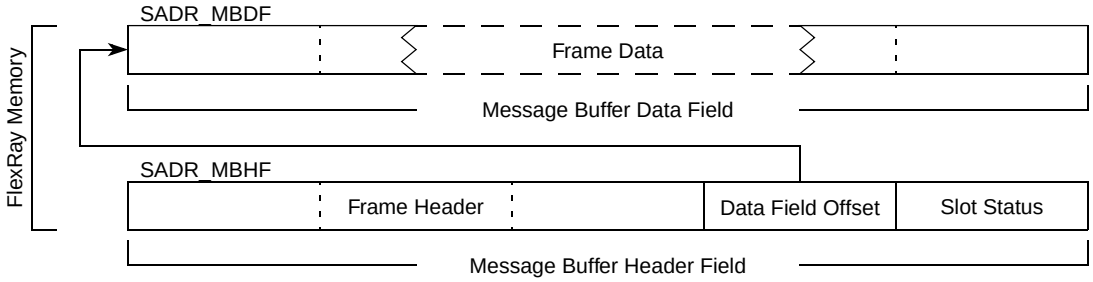


Figure 3-96. Physical Message Buffer Structure

3.4.2.1 Message Buffer Header Field

The message buffer header field is a contiguous region in the FRM and occupies ten bytes. It contains the frame header, the data field offset, and the slot status. Its structure is shown in Figure 3-96. The physical start address *SADR_MBHF* of the message buffer header field must be 16-bit aligned.

3.4.2.1.1 Frame Header

The frame header occupies the first six bytes in the message buffer header field. It contains all FlexRay frame header related information according to the *FlexRay Communications System Protocol*

Specification, Version 2.1 Rev A. A detailed description of the usage and the content of the frame header is provided in Section 3.4.5.2.1, “Frame Header Section Description”.

3.4.2.1.2 Data Field Offset

The data field offset follows the frame header in the message buffer data field and occupies two bytes. It contains the offset of the corresponding message buffer data field with respect to the FlexRay module FRM base address 0x800. The data field offset is used to determine the start address *SADR_MBDF* of the corresponding message buffer data field in the FRM according to Equation 3-1.

SADR_MBDF = [Data Field Offset] + 0x800 *Eqn. 3-1*

3.4.2.1.3 Slot Status

The slot status occupies the last two bytes of the message buffer header field. It provides the slot and frame status related information according to the *FlexRay Communications System Protocol Specification, Version 2.1 Rev A*. A detailed description of the content and usage of the slot status is provided in Section 3.4.5.2.3, “Slot Status Description”.

3.4.2.2 Message Buffer Data Field

The message buffer data field is a contiguous area of 2-byte entities. This field contains the frame payload data, or a part of it, of the frame to be transmitted to or received from the FlexRay bus. The minimum length of this field depends on the specific message buffer configuration and is specified in the message buffer descriptions given in Section 3.4.3, “Message Buffer Types”.

3.4.3 Message Buffer Types

The FlexRay module provides three different types of message buffers.

- Individual Message Buffers
- Receive Shadow Buffers
- Receive FIFO Buffers

For each message buffer type the structure of the physical message buffer is identical. The message buffer types differ only in the structure and content of message buffer control data, which control the related physical message buffer. The message buffer control data are described in the following sections.

3.4.3.1 Individual Message Buffers

The individual message buffers are used for all types of frame transmission and for dedicated frame reception based on individual filter settings for each message buffer. The FlexRay module supports three types of individual message buffers, which are described in Section 3.4.6, “Individual Message Buffer Functional Description”.

Each individual message buffer consists of two parts, the physical message buffer, which is located in the FRM, and the message buffer control data, which are located in dedicated registers. The structure of an individual message buffer is given in Figure 3-97.

Each individual message buffer has a message buffer number n assigned, which determines the set of message buffer control registers associated to this individual message buffer. The individual message buffer with message buffer number *n* is controlled by the registers MBCCSRn, MBCCFRn, MBFIDRn, and MBIDXRn.

The connection between the message buffer control registers and the physical message buffer is established by the message buffer index field MBIDX in the Message Buffer Index Registers (MBIDXRn). The start address SADR_MBHF of the related message buffer header field in the FRM is determined according to Equation 3-2.

$$SADR\_MBHF = (MBIDXRn[MBIDX] * 10) + 0x800 \qquad \textit{Eqn. 3-2}$$

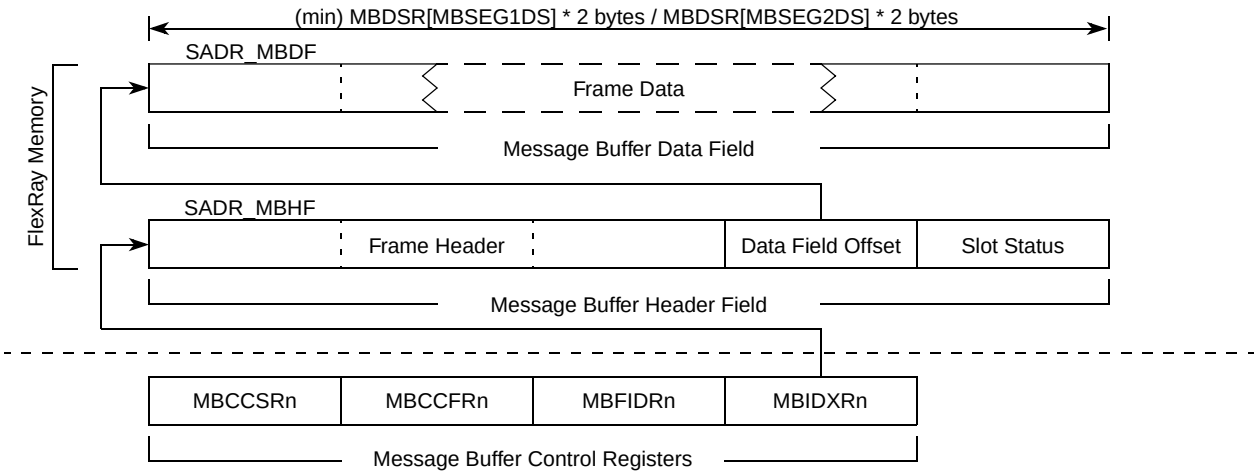


Figure 3-97. Individual Message Buffer Structure

3.4.3.1.1 Individual Message Buffer Segments

The set of the individual message buffers can be split up into two message buffer segments using the Message Buffer Segment Size and Utilization Register (MBSSUTR). All individual message buffers with a message buffer number n less than or equal to MBSSUTR.LAST_MB_SEG1 belong to the first message buffer segment. All individual message buffers with a message buffer number n > MBSSUTR.LAST_MB_SEG1 belong to the second message buffer segment. The following rules apply to the length of the message buffer data field:

- all physical message buffers associated to individual message buffers that belong to the same message buffer segment must have message buffer data fields of the same length
- the minimum length of the message buffer data field for individual message buffers in the first message buffer segment is 2 * MBDSR.MBSEG1DS bytes
- the minimum length of the message buffer data field for individual message buffers assigned to the second segment is 2 * MBDSR.MBSEG2DS bytes.

3.4.3.2 Receive Shadow Buffers

The receive shadow buffers are required for the frame reception process for individual message buffers. The FlexRay module provides four receive shadow buffers, one receive shadow buffer per channel and per message buffer segment.

Each receive shadow buffer consists of two parts, the physical message buffer located in the FRM and the receive shadow buffer control registers located in dedicated registers. The structure of a receive shadow buffer is shown in Figure 3-98. The four internal shadow buffer control registers can be accessed by the Receive Shadow Buffer Index Register (RSBIR).

The connection between the receive shadow buffer control register and the physical message buffer for the selected receive shadow buffer is established by the receive shadow buffer index field RSBIDX in the Receive Shadow Buffer Index Register (RSBIR). The start address SADR_MBHF of the related message buffer header field in the FRM is determined according to Equation 3-3.

$$SADR\_MBHF = (RSBIR[RSBIDX] * 10) + 0x800 \qquad \textit{Eqn. 3-3}$$

The length required for the message buffer data field depends on the message buffer segment that the receive shadow buffer is assigned to. For the receive shadow buffers assigned to the first message buffer segment, the length must be the same as for the individual message buffers assigned to the first message buffer segment. For the receive shadow buffers assigned to the second message buffer segment, the length must be the same as for the individual message buffers assigned to the second message buffer segment. The receive shadow buffer assignment is described in Receive Shadow Buffer Index Register (RSBIR).

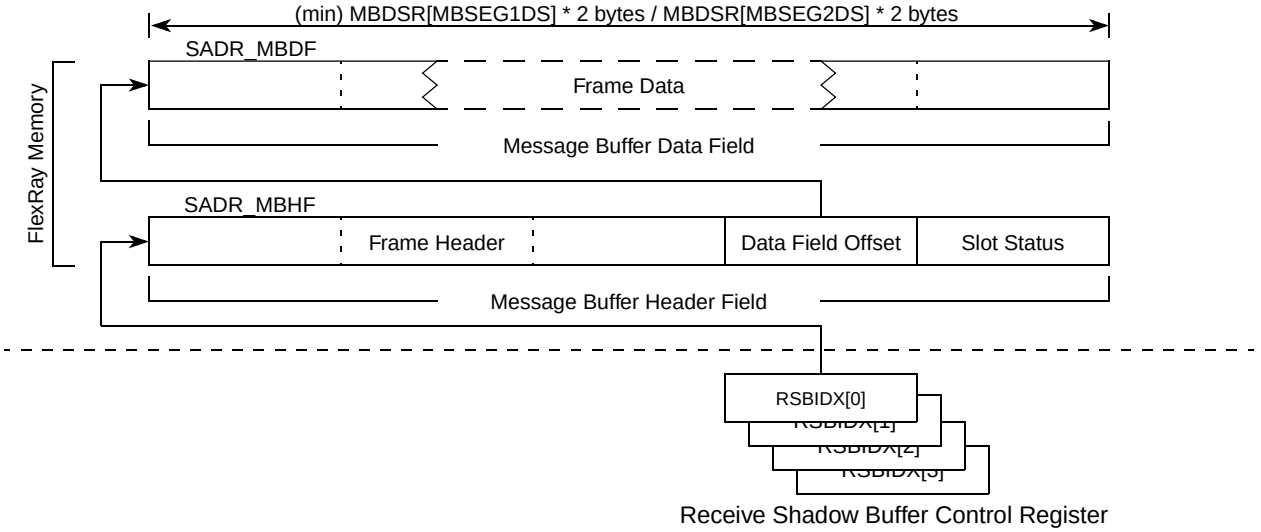


Figure 3-98. Receive Shadow Buffer Structure

3.4.3.3 Receive FIFO

The receive FIFO implements a frame reception system based on the FIFO concept. The FlexRay module provides two independent receive FIFOs, one per channel.

A receive FIFO consists of a set of physical message buffers in the FRM and a set of receive FIFO control registers located in dedicated registers. The structure of a receive FIFO is given in Figure 3-99.

The connection between the receive FIFO control registers and the set of physical message buffers is established by the start index field SIDX in the Receive FIFO Start Index Register (RFSIR), the FIFO depth field FIFO_DEPTH in the Receive FIFO Depth and Size Register (RFDSR), and the read index field RDIDX Receive FIFO A Read Index Register (RFARIR) / Receive FIFO B Read Index Register (RFBRIR). The start address SADR_MBHF_1 of the first message buffer header field that belongs to the receive FIFO in the FRM is determined according to Equation 3-4.

$$SADR\_MBHF[1] = (RFSIR[SIDX] * 10) + 0x800 \qquad \textit{Eqn. 3-4}$$

The start address SADR_MBHF[n] of the last message buffer header field that belongs to the receive FIFO in the FRM is determined according to Equation 3-5.

$$SADR\_MBHF[n] = ((RFSIR[SIDX] + RFDSR[FIFO\_DEPTH]) * 10) + 0x800 \qquad \textit{Eqn. 3-5}$$

NOTE

All message buffer header fields assigned to a receive FIFO must be a contiguous region.

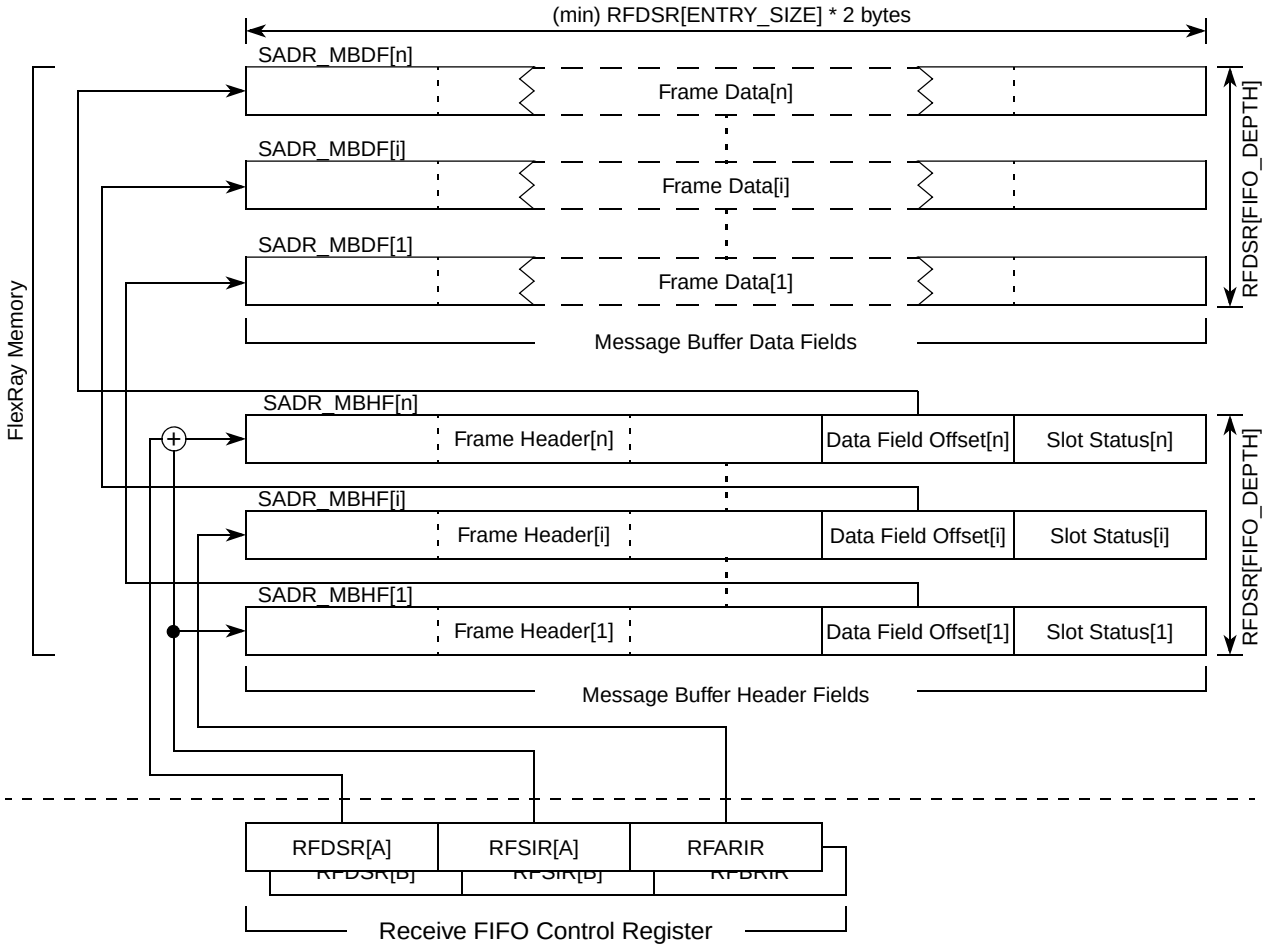


Figure 3-99. Receive FIFO Structure

3.4.3.4 Message Buffer Configuration and Control Data

This section describes the configuration and control data for each message buffer type.

3.4.3.4.1 Individual Message Buffer Configuration Data

Before an individual message buffer can be used for transmission or reception, it must be configured. There is a set of common configuration parameters that applies to all individual message buffers and a set of configuration parameters that applies to each message buffer individually.

Common Configuration Data

The set of common configuration data for individual message buffers is located in the following registers.

- Message Buffer Data Size Register (MBDSR)
  The MBSEG2DS and MBSEG1DS fields define the minimum length of the message buffer data field with respect to the message buffer segment.
- Message Buffer Segment Size and Utilization Register (MBSSUTR)
  The LAST_MB_SEG1 and LAST_MB_UTIL fields define the segmentation of the individual message buffers and the number of individual message buffers that are used. For more details, see Section 3.4.3.1.1, “Individual Message Buffer Segments”

Specific Configuration Data

The set of message buffer specific configuration data for individual message buffers is located in the following registers.

- Message Buffer Configuration, Control, Status Registers (MBCCSRn)
  The MCM, MBT, MTD bits configure the message buffer type.
- Message Buffer Cycle Counter Filter Registers (MBCCFRn)
  The MTM, CHA, CHB bits configure the transmission mode and the channel assignment. The CCFE, CCFMSK, and CCFVAL bits and fields configure the cycle counter filter.
- Message Buffer Frame ID Registers (MBFIDRn)
  For a transmit message buffer, the FID field is used to determine the slot in which the message in this message buffer is transmitted.
- Message Buffer Index Registers (MBIDXRn)
  This MBIDX field provides the index of the message buffer header field of the physical message buffer that is currently associated with this message buffer.

3.4.3.5 Individual Message Buffer Control Data

During normal operation, each individual message buffer can be controlled by the control and trigger bits CMT, LCKT, EDT, and MBIE in the Message Buffer Configuration, Control, Status Registers (MBCCSRn).

3.4.3.6 Receive Shadow Buffer Configuration Data

Before frame reception into the individual message buffers can be performed, the receive shadow buffers must be configured. The configuration data are provided by the Receive Shadow Buffer Index Register (RSBIR). For each receive shadow buffer, the application provides the message buffer header index. When the protocol is in the *POC:normal active* or *POC:normal passive* state, the receive shadow buffers are under full FlexRay module control.

3.4.3.7 Receive FIFO Control and Configuration Data

This section describes the configuration and control data for the two receive FIFOs.

3.4.3.7.1 Receive FIFO Configuration Data

The FlexRay module provides two completely independent receive FIFOs, one per channel. Each FIFO has its own set of configuration data. The configuration data are located in the following registers:

- Receive FIFO Start Index Register (RFSIR)
- Receive FIFO Depth and Size Register (RFDSR)
- Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR)
- Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR)
- Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR)
- Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR)
- Receive FIFO Range Filter Configuration Register (RFRFCFR)

3.4.3.7.2 Receive FIFO Control Data

The application can access the receive FIFO at any time using the values provided in the Receive FIFO A Read Index Register (RFARIR) and Receive FIFO B Read Index Register (RFBRIR). To update the Receive FIFO A Read Index Register (RFARIR), the application must write 1 to the FIFO A Not Empty Interrupt Flag FNEAIF in the Global Interrupt Flag and Enable Register (GIFER). To update the Receive FIFO B Read Index Register (RFBRIR) the application must write 1 to the FIFO B Not Empty Interrupt Flag FNEBIF in the Global Interrupt Flag and Enable Register (GIFER). As long as the FIFO is not empty, each update increments the related read index. If the read index has reached the last FIFO entry, it wraps back to the FIFO start index.

3.4.4 FlexRay Memory Layout

The FlexRay module supports a wide range of possible layouts for the FRM. Figure 3-100 shows an example layout. The following set of rules applies to the layout of the FRM:

- The FRM is a contiguous region.
- The FRM size is 6 Kbytes.
- The FRM starts at modue address 0x800.

The FRM contains three areas: the *message buffer header area*, the *message buffer data area*, and the *sync frame table area*. The areas are described in this section.

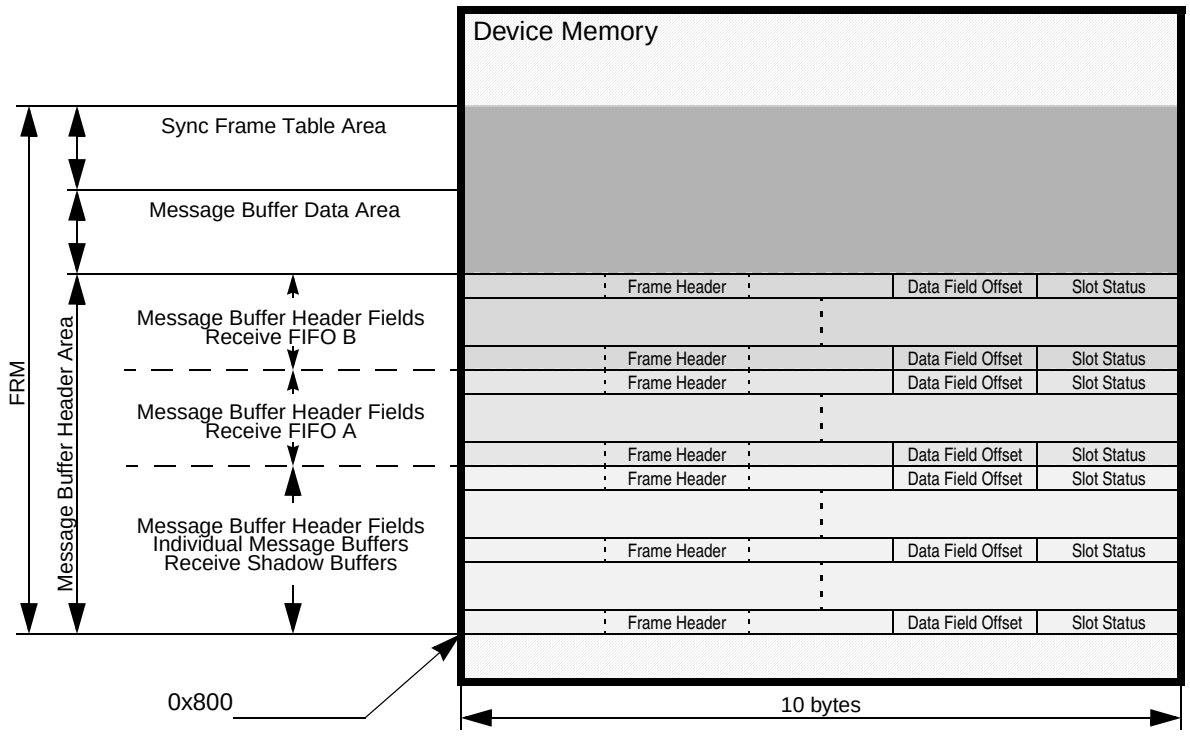


Figure 3-100. Example of FRM Layout

3.4.4.1 Message Buffer Header Area

The message buffer header area contains all message buffer header fields of the physical message buffers for all message buffer types. The following rules apply to the message buffer header fields for the three type of message buffers.

1. The start address SADR_MBHF of each message buffer header field for *individual message buffers* and *receive shadow buffers* must fulfill Equation 3-6.

$$SADR\_MBHF = (i * 10) + 0x800; (0 <= i < 256) \quad \textit{Eqn. 3-6}$$

2. The start address SADR_MBHF of each message buffer header field for the *receive FIFO* must fulfill Equation 3-7.

$$SADR\_MBHF = (i * 10) + 0x800; (0 <= i < 256) \quad \textit{Eqn. 3-7}$$

3. The message buffer header fields for a receive FIFO have to be a contiguous area.

3.4.4.2 Message Buffer Data Area

The message buffer data area contains all the message buffer data fields of the physical message buffers. Each message buffer data field must start at a 16-bit boundary.

3.4.4.3 Sync Frame Table Area

The sync frame table area is used to provide a copy of the internal sync frame tables for application access. Refer to Section 3.4.12, “Sync Frame ID and Sync Frame Deviation Tables” for the description of the sync frame table area.

3.4.5 Physical Message Buffer Description

This section provides a detailed description of the usage and the content of the two parts of a physical message buffer, the message buffer header field and the message buffer data field.

3.4.5.1 Message Buffer Protection and Data Consistency

The physical message buffers are located in the FRM. The FlexRay module provides no means to protect the FRM from uncontrolled or illegal host or other client write access. To ensure data consistency of the physical message buffers, the application must follow the write access scheme that is given in the description of each of the physical message buffer fields.

3.4.5.2 Message Buffer Header Field Description

This section provides a detailed description of the usage and content of the message buffer header field. A description of the structure of the message buffer header fields is given in Section 3.4.2.1, “Message Buffer Header Field”. Each message buffer header field consists of three sections: the frame header section, the data field offset, and the slot status section. For a detailed description of the Data Field Offset, see Section 3.4.2.1.2, “Data Field Offset”.

3.4.5.2.1 Frame Header Section Description

Frame Header Section Content

The semantic and content of the frame header section depends on the message buffer type.

For individual receive message buffers and receive FIFOs, the frame header receives the frame header data of the *first valid frame* received on the assigned channels. If a receive message buffer is assigned to both channels, the first valid frame received on channel A or channel B is stored.

For receive shadow buffers, the frame header receives the frame header data of the current frame received regardless of whether the frame is valid or not.

For single and double transmit message buffers, the application writes the frame header of the frame to be transmitted into this location. The frame header is read out when the frame is transferred to the FlexRay bus.

The structure of the frame header in the message buffer header field is given in Figure 3-101. A detailed description of the frame header fields is given in Table 3-81.

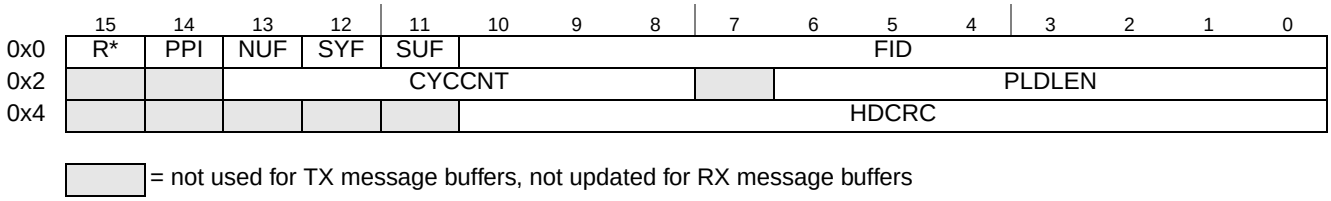


Figure 3-101. Frame Header Structure

Frame Header Section Access

The frame header is located in the FRM. To ensure data consistency, the application must follow the write access scheme described below.

For receive message buffers, receive shadow buffers, and receive FIFOs, the application must not write to the frame header field.

For transmit message buffers, the application must follow the write access restrictions given in Table 3-80. This table shows the condition under which the application can write to the frame header entries. In general, the application can modify all frame header entries when the protocol is in the *POC:config* state or when the message buffer is disabled. For message buffers assigned to the dynamic segment, the application can modify all frame header entries except the frame ID when the message buffer is locked.

Table 3-80. Frame Header Write Access Constraints

Field	TX					
	Single Buffered		Double Buffered			
	Static Segment	Dynamic Segment	Static Segment		Dynamic Segment	
			Commit Side	Transmit Side	Commit Side	Transmit Side
FID	*POC:config* or MB_DIS					
R*, PPI NUF, SYF SUF CYCCNT PLDLEN HDCRD	*POC:config* or MB_DIS	*POC:config* or MB_DIS or MB_LCK	*POC:config* or MB_DIS		*POC:config* or MB_DIS or MB_LCK	*POC:config* or MB_DIS

The frame header entries NUF, SYF, SUF, and CYCCNT are not used for frame transmission. These values are generated internally before frame transmission depending on the current transmission state and configuration.

For transmit message buffers assigned to the *static* segment, the PLDLEN value must be equal to the value of the payload_length_static field in the Protocol Configuration Register 19 (PCR19). If this is not fulfilled, the static payload length error flag SPL_EF in the CHI Error Flag Register (CHIERFR) is set when the message buffer is under transmission. The PE generates a syntactically and semantically correct frame with payload_length_static payload words and the payload length field in the frame header set to payload_length_static.

For transmit message buffers assigned to the *dynamic* segment, the PLDLEN value must be less than or equal to the value of the max_payload_length_dynamic field in the Protocol Configuration Register 24 (PCR24). If this is not fulfilled, the dynamic payload length error flag DPL_EF in the CHI Error Flag Register (CHIERFR) is set when the message buffer is under transmission. The PE generates a syntactically and semantically correct dynamic frame with PLDLEN payload words and the payload length field in the frame header set to PLDLEN.

Table 3-81. Frame Header Field Descriptions

Field	Description
R*	**Reserved Bit** — This bit corresponds to the *Reserved bit* in the FlexRay frame header. • For receive and FIFO message buffers, this is a status bit and represents the value of the Reserved bit in the frame received on the FlexRay bus in the corresponding slot. • For transmit message buffers, this is a control bit. The FlexRay module transmits this within the frame header. **Note:** For protocol compliant operation, this control bit must be set to 0 for transmit message buffers.
PPI	**Payload Preamble Indicator** — This bit corresponds to the *Payload Preamble Indicator* in the FlexRay frame header. • For receive and FIFO message buffers, this is a status bit and represents the value of the *Payload Preamble Indicator* of the first valid frame received on the FlexRay in the slot indicated by the CYCCNT field. • For transmit message buffers, this is a control bit. The FlexRay module uses this value to set the *Payload Preamble Indicator* in the frame header of the frame to transmit. 0 No network management vector or message ID in frame payload data 1 Static Segment: Frame payload data contains network management vector Dynamic Segment: Frame payload data contains message ID
NUF	**Null Frame Indicator** — This bit corresponds to the *Null Frame Indicator* in the FlexRay frame header. • For receive message buffers and receive FIFOs, this is a status bit and represents the value of the *Null Frame Indicator* of the first valid frame received on the FlexRay bus in the slot indicated by the CYCCNT field. • For transmit message buffers, the value of this bit is ignored. The FlexRay module determines internally whether a null frame or non-null frame must be transmitted and sets the *Null Frame Indicator* accordingly. 0 Null frame received 1 Normal frame received
SYF	**Sync Frame Indicator** — This bit corresponds to the *Sync Frame Indicator* in the FlexRay frame header. • For receive message buffers and receive FIFOs, this is a status bit and represents the value of the *Sync Frame Indicator* of the first valid frame received on the FlexRay bus in the slot indicated by the CYCCNT field. • For transmit message buffers, the value of this bit is ignored. The FlexRay module determines internally whether a sync frame must be transmitted and sets the *Sync Frame Indicator* accordingly.
SUF	**Startup Frame Indicator** — This bit corresponds to the *Startup Frame Indicator* in the FlexRay frame header. • For receive message buffers and receive FIFOs, this is a status bit and represents the value of the *Startup Frame Indicator* of the first valid frame received on the FlexRay bus in the slot indicated by the CYCCNT field • For transmit message buffers, the value of this bit is ignored. The FlexRay module determines internally whether a startup frame must be transmitted and sets the *Startup Frame Indicator* accordingly.
FID	**Frame ID** • For receive message buffers and receive FIFOs, this field provides the value of the *Frame ID* field of the first valid frame received on the FlexRay bus in the slot indicated by the CYCCNT field. • For transmit message buffers, this field provides the value transmitted in the *Frame ID* field of the FlexRay frame. **Note:** For transmit message buffers, the application must program this field to the same value as in the corresponding Message Buffer Frame ID Registers (MBFIDRn). If the FlexRay module detects a mismatch while transmitting the frame header, it sets the frame ID error flag FID_EF in the CHI Error Flag Register (CHIERFR). The value of the FID field is ignored and replaced by the value provided in the Message Buffer Frame ID Registers (MBFIDRn).

Table 3-81. Frame Header Field Descriptions (Continued)

Field	Description
CYCCNT	**Cycle Count** • For receive message buffer and receive FIFOs, this field provides the number of the communication cycle in which the frame stored in this message buffer was received. • For transmit message buffers, the value of this field is ignored. The FlexRay module overwrites this value with the current cycle count value when it transmits the frame.
PLDLEN	**Payload Length in 16-Bit Units** • For receive message buffers and receive FIFOs, this field provides the value of the payload length field of the first valid frame received on the FlexRay bus in the slot indicated by the FID field. • For transmit message buffers assigned to the static segment, this value is ignored for the frame generation. The FlexRay module uses the value in the PCR19.paylaod_length_static to set the value of the *Payload length* field in the transmitted frame. • For transmit message buffers assigned to the dynamic segment, this value is used to set the value of the *Payload length* field in the transmitted frame. **Note:** The value of this field is given in numbers of 16-bit units
HDCRC	**Header CRC** • For receive and FIFO message buffers, this field provides the value of the *Header CRC* of the received frame. • For transmit message buffers, this field provides the *Header CRC* value as it was given by the application.The FlexRay module transmits this value in the *Header CRC* field of the transmitted frame.

3.4.5.2.2 Data Field Offset Description

Data Field Offset Content

For a detailed description of the Data Field Offset, see Section 3.4.2.1.2, “Data Field Offset”.

Data Field Offset Access

The application shall program the Data Field Offset when configuring the message buffers in the *POC:config* state or when the message buffer is disabled.

3.4.5.2.3 Slot Status Description

The slot status is a read-only structure for the application and a write-only structure for the FlexRay module. The meaning and content of the slot status in the message buffer header field depends on the message buffer type.

Receive Message Buffer and Receive FIFO Slot Status Description

This section describes the slot status structure for the individual receive message buffers and receive FIFOs. The content of the slot status structure for receive message buffers depends on the message buffer type and on the channel assignment for individual receive message buffers as given by Table 3-82.

Table 3-82. Receive Message Buffer Slot Status Content

Receive Message Buffer Type	Slot Status Content
Individual Receive Message Buffer assigned to both channels MBCCSRn.CHA = 1 and MBCCSRn.CHB = 1	see Figure 3-102
Individual Receive Message Buffer assigned to channel A MBCCSRn.CHA = 1 and MBCCSRn.CHB = 0	see Figure 3-103
Individual Receive Message Buffer assigned to channel B MBCCSRn.CHA = 0 and MBCCSRn.CHB = 1	see Figure 3-104
Receive FIFO Channel A Message Buffer	see Figure 3-103
Receive FIFO Channel B Message Buffer	see Figure 3-104

The meaning of the bits in the slot status structure is explained in Table 3-83.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFB	SYB	NFB	SUB	SEB	CEB	BVB	CH	VFA	SYA	NFA	SUA	SEA	CEA	BVA	0
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-102. Receive Message Buffer Slot Status Structure (ChAB)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	VFA	SYA	NFA	SUA	SEA	CEA	BVA	0
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-103. Receive Message Buffer Slot Status Structure (ChA)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFB	SYB	NFB	SUB	SEB	CEB	BVB	1	0	0	0	0	0	0	0	0
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-104. Receive Message Buffer Slot Status Structure (ChB)

Table 3-83. Receive Message Buffer Slot Status Field Descriptions

Field	Description
Common Message Buffer Status Bits	
15 VFB	**Valid Frame on Channel B** — protocol related variable: *vSS!ValidFrame* channel B 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
14 SYB	**Sync Frame Indicator Channel B** — protocol related variable: *vRF!Header!SyFIndicator* channel B 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
13 NFB	**Null Frame Indicator Channel B** — protocol related variable: *vRF!Header!NFIndicator* channel B 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
12 SUB	**Startup Frame Indicator Channel B** — protocol related variable: *vRF!Header!SuFIndicator* channel B 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1

Table 3-83. Receive Message Buffer Slot Status Field Descriptions (Continued)

Field	Description
11 SEB	**Syntax Error on Channel B** — protocol related variable: *vSS!SyntaxError* channel B 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
10 CEB	**Content Error on Channel B** — protocol related variable: *vSS!ContentError* channel B 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1
9 BVB	**Boundary Violation on Channel B** — protocol related variable: *vSS!BViolation* channel B 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1
8 CH	**Channel first valid received** — This status bit applies only to receive message buffers assigned to the static segment and to both channels. It indicates the channel that has received the *first valid* frame in the slot. This flag is set to 0 if no valid frame was received at all in the subscribed slot. 0 first valid frame received on channel A, or no valid frame received at all 0 first valid frame received on channel B
7 VFA	**Valid Frame on Channel A** — protocol related variable: *vSS!ValidFrame* channel A 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
6 SYA	**Sync Frame Indicator Channel A** — protocol related variable: *vRF!Header!SyFIndicator* channel A 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
5 NFA	**Null Frame Indicator Channel A** — protocol related variable: *vRF!Header!NFIndicator* channel A 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
4 SUA	**Startup Frame Indicator Channel A** — protocol related variable: *vRF!Header!SuFIndicator* channel A 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1
3 SEA	**Syntax Error on Channel A** — protocol related variable: *vSS!SyntaxError* channel A 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
2 CEA	**Content Error on Channel A** — protocol related variable: *vSS!ContentError* channel A 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1
1 BVA	**Boundary Violation on Channel A** — protocol related variable: *vSS!BViolation* channel A 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1

Transmit Message Buffer Slot Status Description

This section describes the slot status structure for transmit message buffers. Only the TCA and TCB status bits are directly related to the transmission process. All other status bits in this structure are related to a receive process that may have occurred. The content of the slot status structure for transmit message buffers depends on the channel assignment as given by Table 3-84.

Table 3-84. Transmit Message Buffer Slot Status Content

Transmit Message Buffer Type	Slot Status Content
Individual Transmit Message Buffer assigned to both channels MBCCSRn.CHA = 1 and MBCCSRn.CHB = 1	see Figure 3-105
Individual Transmit Message Buffer assigned to channel A MBCCSRn.CHA = 1 and MBCCSRn.CHB = 0	see Figure 3-106
Individual Transmit Message Buffer assigned to channel B MBCCSRn.CHA = 0 and MBCCSRn.CHB = 1	see Figure 3-107

The meaning of the bits in the slot status structure is described in Table 3-83.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFB	SYB	NFB	SUB	SEB	CEB	BVB	TCB	VFA	SYA	NFA	SUA	SEA	CEA	BVA	TCA
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-105. Transmit Message Buffer Slot Status Structure (ChAB)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	VFA	SYA	NFA	SUA	SEA	CEA	BVA	TCA
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-106. Transmit Message Buffer Slot Status Structure (ChA)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VFB	SYB	NFB	SUB	SEB	CEB	BVB	TCB	0	0	0	0	0	0	0	0
Reset	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–

Figure 3-107. Transmit Message Buffer Slot Status Structure (ChB)

Table 3-85. Transmit Message Buffer Slot Status Structure Field Descriptions

Field	Description
15 VFB	**Valid Frame on Channel B** — protocol related variable: *vSS!ValidFrame* channel B 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
14 SYB	**Sync Frame Indicator Channel B** — protocol related variable: *vRF!Header!SyFIndicator* channel B 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
13 NFB	**Null Frame Indicator Channel B** — protocol related variable: *vRF!Header!NFIndicator* channel B 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
12 SUB	**Startup Frame Indicator Channel B** — protocol related variable: *vRF!Header!SuFIndicator* channel B 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1
11 SEB	**Syntax Error on Channel B** — protocol related variable: *vSS!SyntaxError* channel B 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
10 CEB	**Content Error on Channel B** — protocol related variable: *vSS!ContentError* channel B 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1

Table 3-85. Transmit Message Buffer Slot Status Structure Field Descriptions (Continued)

Field	Description
9 BVB	**Boundary Violation on Channel B** — protocol related variable: *vSS!BViolation* channel B 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1
8 TCB	**Transmission Conflict on Channel B** — protocol related variable: *vSS!TxConflict* channel B 0 *vSS!TxConflict* = 0 1 *vSS!TxConflict* = 1
7 VFA	**Valid Frame on Channel A** — protocol related variable: *vSS!ValidFrame* channel A 0 *vSS!ValidFrame* = 0 1 *vSS!ValidFrame* = 1
6 SYA	**Sync Frame Indicator Channel A** — protocol related variable: *vRF!Header!SyFIndicator* channel A 0 *vRF!Header!SyFIndicator* = 0 1 *vRF!Header!SyFIndicator* = 1
5 NFA	**Null Frame Indicator Channel A** — protocol related variable: *vRF!Header!NFIndicator* channel A 0 *vRF!Header!NFIndicator* = 0 1 *vRF!Header!NFIndicator* = 1
4 SUA	**Startup Frame Indicator Channel A** — protocol related variable: *vRF!Header!SuFIndicator* channel A 0 *vRF!Header!SuFIndicator* = 0 1 *vRF!Header!SuFIndicator* = 1
3 SEA	**Syntax Error on Channel A** — protocol related variable: *vSS!SyntaxError* channel A 0 *vSS!SyntaxError* = 0 1 *vSS!SyntaxError* = 1
2 CEA	**Content Error on Channel A** — protocol related variable: *vSS!ContentError* channel A 0 *vSS!ContentError* = 0 1 *vSS!ContentError* = 1
1 BVA	**Boundary Violation on Channel A** — protocol related variable: *vSS!BViolation* channel A 0 *vSS!BViolation* = 0 1 *vSS!BViolation* = 1
0 TCA	**Transmission Conflict on Channel A** — protocol related variable: *vSS!TxConflict* channel A 0 *vSS!TxConflict* = 0 1 *vSS!TxConflict* = 1

3.4.5.3 Message Buffer Data Field Description

The message buffer data field is used to store the frame payload data, or a part of it, of the frame to be transmitted to or received from the FlexRay bus. The minimum required length of this field depends on the message buffer type that the physical message buffer is assigned to and is given in Table 3-86. The structure of the message buffer data field is given in Figure 3-108.

Table 3-86. Message Buffer Data Field Minimum Length

physical message buffer assigned to	minimum length defined by
Individual Message Buffer in Segment 1	MBDSR.MBSEG1DS
Receive Shadow Buffer in Segment 1	MBDSR.MBSEG1DS
Individual Message Buffer in Segment 2	MBDSR.MBSEG2DS
Receive Shadow Buffer in Segment 2	MBDSR.MBSEG2DS
Receive FIFO for channel A	RFDSR.ENTRY_SIZE (RFSR.SEL = 0)

Table 3-86. Message Buffer Data Field Minimum Length

physical message buffer assigned to	minimum length defined by
Receive FIFO for channel B	RFDSR.ENTRY_SIZE (RFSR.SEL = 1)

NOTE

The FlexRay module does not access any locations outside the message buffer data field boundaries given by Table 3-86.

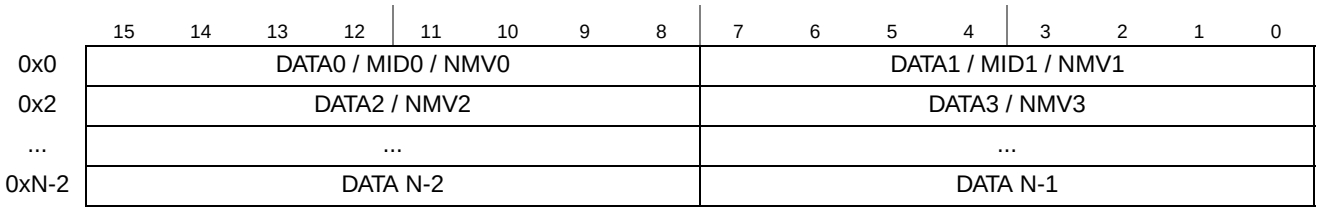


Figure 3-108. Message Buffer Data Field Structure

The message buffer data field is located in the FRM; thus, the FlexRay module has no means to control application write access to the field. To ensure data consistency, the application must follow a write and read access scheme.

3.4.5.3.1 Message Buffer Data Field Read Access

For transmit message buffers, the FlexRay module does not modify the content of the Message Buffer Data Field. Thus the application can read back the data at any time without any impact on data consistency.

For receive message buffers the application must lock the related receive message buffer and retrieve the message buffer header index from the Message Buffer Index Registers (MBIDXRn). While the message buffer is locked, the FlexRay module does not update the Message Buffer Data Field.

For receive FIFOs, the application can read the message buffer indicated by the Receive FIFO A Read Index Register (RFARIR) or the Receive FIFO B Read Index Register (RFBRIR) when the related receive FIFO non-empty interrupt flag FNEAIF or FNEBIF is set in the Global Interrupt Flag and Enable Register (GIFER). While the non-empty interrupt flag is set, the FlexRay module does not update the Message Buffer Data Field related to message buffer indicated by Receive FIFO A Read Index Register (RFARIR) or the Receive FIFO B Read Index Register (RFBRIR).

3.4.5.3.2 Message Buffer Data Field Write Access

For receive message buffers, receive shadow buffers, and receive FIFOs, the application must not write to the message buffer data field.

For transmit message buffers, the application must follow the write access restrictions given in Table 3-87.

Table 3-87. Frame Data Write Access Constraints

Field	single buffered	double buffered	
		commit side	transmit side
DATA, MID, NMV	*POC:config* or MB_DIS or MB_LCK	*POC:config* or MB_DIS or MB_LCK	*POC:config* or MB_DIS

Table 3-88. Frame Data Field Descriptions

Field	Description
DATA[0:N-1]	**Message Data** — Provides the message data received or to be transmitted. For receive message buffer and receive FIFOs, this field provides the message data received for this message buffer. For transmit message buffers, the field provides the message data to be transmitted.
MID[0:1]	**Message Identifier** — If the payload preamble bit PPI is set in the message buffer frame header, the MID field holds the message ID of a dynamic frame located in the message buffer. The receive FIFO filter uses the received message ID for message ID filtering.
NMV[0:11]	**Network Management Vector** — If the payload preamble bit PPI is set in the message buffer frame header, the network management vector field holds the network management vector of a static frame located in the message buffer. **Note:** The MID and NMV bytes replace the corresponding DATA bytes.

3.4.6 Individual Message Buffer Functional Description

The FlexRay module provides three basic types of individual message buffers:

1. Single Transmit Message Buffers
2. Double Transmit Message Buffers
3. Receive Message Buffers

Before an individual message buffer can be used, it must be configured by the application. After the initial configuration, the message buffer can be reconfigured later. The set of the configuration data for individual message buffers is given in Section 3.4.3.4.1, “Individual Message Buffer Configuration Data”.

3.4.6.1 Individual Message Buffer Configuration

The individual message buffer configuration consists of two steps. The first step is the allocation of the required amount of memory for the FRM. The second step is the programming of the message buffer configuration registers, which is described in this section.

3.4.6.1.1 Common Configuration Data

One part of the message buffer configuration data is common to all individual message buffers and the receive shadow buffers. These data can only be set when the protocol is in the *POC:config* state.

The application configures the number of utilized individual message buffers by writing the message buffer number of the last utilized message buffer into the LAST_MB_UTIL field in the Message Buffer Segment Size and Utilization Register (MBSSUTR).

The application configures the size of the two segments of individual message buffers by writing the message buffer number of the last message buffer in the first segment into the LAST_MB_SEG1 field in the Message Buffer Segment Size and Utilization Register (MBSSUTR)

The application configures the length of the message buffer data fields for both of the message buffer segments by writing to the MBSEG2DS and MBSEG1DS fields in the Message Buffer Data Size Register (MBDSR).

Depending on the current receive functionality of the FlexRay module, the application must configure the receive shadow buffers. For each segment and for each channel with at least one individual receive message buffer assigned, the application must configure the related receive shadow buffer using the Receive Shadow Buffer Index Register (RSBIR).

3.4.6.1.2 Specific Configuration Data

The second part of the message buffer configuration data is specific for each message buffer.

This data can be changed only when either of the following two events occur:

- The protocol is in the*POC:config* state
- The message buffer is disabled, MBCCSRn.EDS equals 0

The individual message buffer type is defined by the MTD and MBT bits in the Message Buffer Configuration, Control, Status Registers (MBCCSRn) as given in Table 3-89.

Table 3-89. Individual Message Buffer Types

MBCCSRn.MTD	MBCCSRn.MBT	Individual Message Buffer Description
0	0	Receive Message Buffer
0	1	Reserved
1	0	Single Transmit Message Buffer
1	1	Double Transmit Message Buffer

The message buffer specific configuration data are

1. MCM, MBT, MTD bits in Message Buffer Configuration, Control, Status Registers (MBCCSRn)
2. all fields and bits in Message Buffer Cycle Counter Filter Registers (MBCCFRn)
3. all fields and bits in Message Buffer Frame ID Registers (MBFIDRn)
4. all fields and bits in Message Buffer Index Registers (MBIDXRn)

The meaning of the specific configuration data depends on the message buffer type, as given in the detailed message buffer type descriptions Section 3.4.6.2, “Single Transmit Message Buffers”, Section 3.4.6.3, “Receive Message Buffers”, and Section 3.4.6.4, “Double Transmit Message Buffer”.

3.4.6.2 Single Transmit Message Buffers

The section provides a detailed description of the functionality of single buffered transmit message buffers.

A single transmit message buffer is used by the application to provide message data to the FlexRay module transmitted over the FlexRay Bus. The FlexRay module uses the transmit message buffers to provide

information about the transmission process and status information about the slot in which message was transmitted.

The individual message buffer with message buffer number *n* is configured to be a single transmit message buffer by the following settings:

- MBCCSRn.MBT equals 0 (single buffered message buffer)
- MBCCSRn.MTD equals 1 (transmit message buffer)

3.4.6.2.1 Access Regions

To certain message buffer fields, both the application and the FlexRay module have access. To ensure data consistency, a message buffer locking scheme is implemented, which is used to control the access to the data, control, and status bits of a message buffer. The access regions for single transmit message buffers are depicted in Figure 3-109. A description of the regions is given in Table 3-90. If an region is active as indicated in Table 3-91, the access scheme given for that region applies to the message buffer.

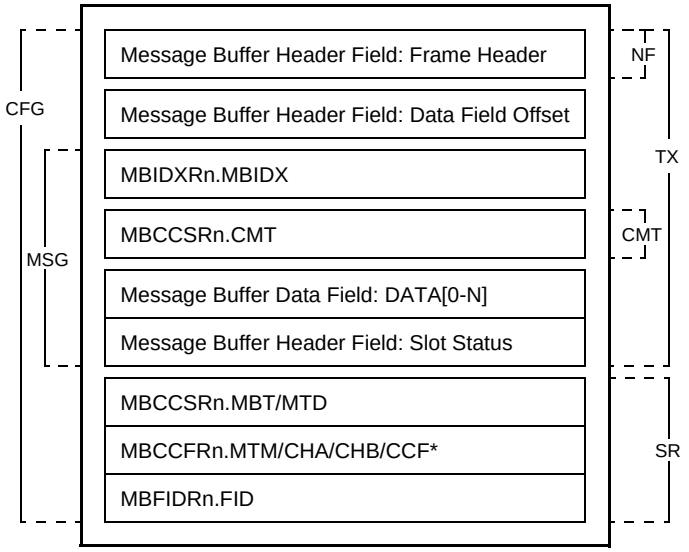


Figure 3-109. Single Transmit Message Buffer Access Regions

Table 3-90. Single Transmit Message Buffer Access Regions Description

Region	Access from		Region used for
	Application	Module	
CFG	read/write	-	Message Buffer Configuration
MSG	read/write	-	Message Data and Slot Status Access
NF	-	read-only	Message Header Access for Null Frame Transmission
TX	-	read/write	Message Transmission and Slot Status Update
CM	-	read-only	Message Buffer Validation
SR	-	read-only	Message Buffer Search

The trigger bits MBCCSRn.EDT and MBCCSRn.LCKT, and the interrupt enable bit MBCCSRn.MBIE are not under access control and can be accessed from the application at any time. The status bits

MBCCSRn.EDS and MBCCSRn.LCKS are not under access control and can be accessed from the FlexRay module at any time.

The interrupt flag MBCCSnR.MBIF is not under access control and can be accessed from the application and the FlexRay module at any time. FlexRay module clear access has higher priority.

The FlexRay module restricts its access to the regions depending on the current state of the message buffer. The application must adhere to these restrictions to ensure data consistency. The transmit message buffer states are given in Figure 3-110. A description of the states is given in Table 3-91, which also provides the access scheme for the access regions.

The status bits MBCCSRn.EDS and MBCCSRn.LCKS provide the application with the required message buffer status information. The internal status information is not visible to the application.

3.4.6.2.2 Message Buffer States

This section describes the transmit message buffer states and provides a state diagram.

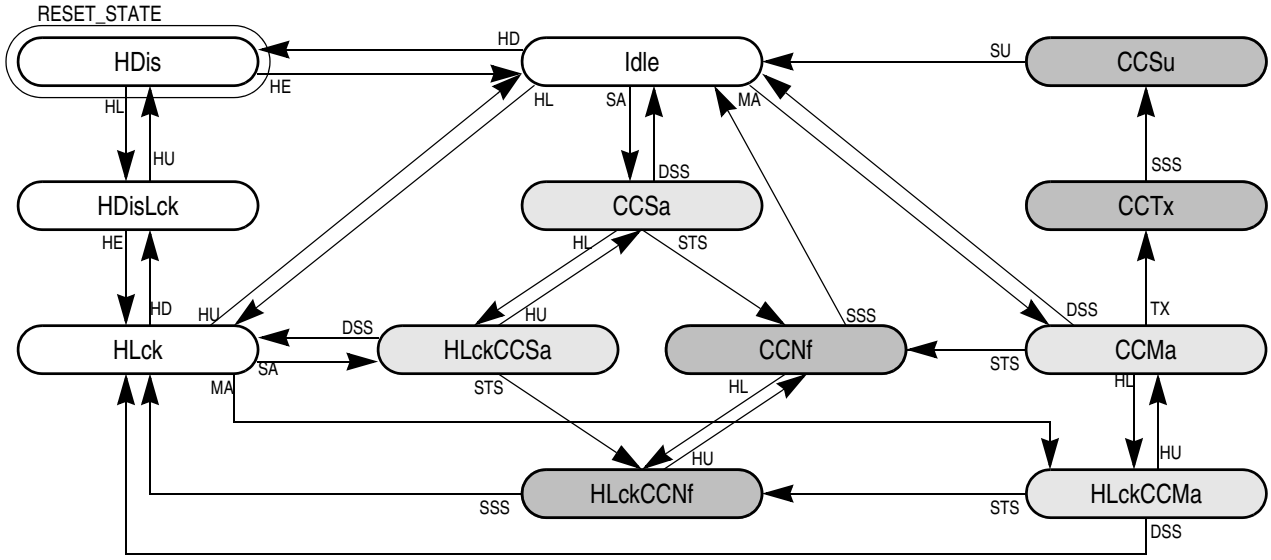


Figure 3-110. Single Transmit Message Buffer States

Table 3-91. Single Transmit Message Buffer State Description

State	MBCCSRn		Access Region		Description
	EDS	LCKS	Appl.	Module	
Idle	1	0	–	CM, SR	**Idle** - Message Buffer is idle. Included in message buffer search.
HDis	0	0	CFG	–	**Disabled** - Message Buffer under configuration. Excluded from message buffer search.
HDisLck	0	1	CFG	–	**Disabled and Locked** - Message Buffer under configuration. Excluded from message buffer search.
HLck	1	1	MSG	SR	**Locked** - Applications access to data, control, and status. Included in message buffer search.

Table 3-91. Single Transmit Message Buffer State Description (Continued)

State	MBCCSRn		Access Region		Description
	EDS	LCKS	Appl.	Module	
CCSa	1	0	–	–	**Slot Assigned** - Message buffer assigned to next static slot. Ready for Null Frame transmission.
HLckCCSa	1	1	MSG	–	**Locked and Slot Assigned** - Applications access to data, control, and status.Message buffer assigned to next static slot
CCNf	1	0	–	NF	**Null Frame Transmission** Header is used for null frame transmission.
HLckCCNf	1	1	MSG	NF	**Locked and Null Frame Transmission** - Applications access to data, control, and status. Header is used for null frame transmission.
CCMa	1	0	–	CM	**Message Available** - Message buffer is assigned to next slot and cycle counter filter matches.
HLckCCMa	1	1	MSG	–	**Locked and Message Available** - Applications access to data, control, and status. Message buffer is assigned to next slot and cycle counter filter matches.
CCTx	1	0	–	TX	**Message Transmission** - Message buffer data transmit. Payload data from buffer transmitted
CCSu	1	0	–	TX	**Status Update** - Message buffer status update. Update of status flags, the slot status field, and the header index.

3.4.6.2.3 Message Buffer Transitions

Application Transitions

The application transitions can be triggered by the application using the commands described in Table 3-92. The application issues the commands by writing to the Message Buffer Configuration, Control, Status Registers (MBCCSRn). Only one command can be issued with one write access. Each command is executed immediately. If the command is ignored, it must be issued again.

The enable and disable commands issued by writing 1 to the trigger bit MBCCSRn.EDT. The transition that triggered by each of these command depends on the current value of the status bit MBCCSRn.EDS. If the command triggers the disable transition *HD* and the message buffer is in one of the states *CCSa*, *HLckCCSa*, *CCMa*, *HLckCCMa, CCNf*, *HLckCCNf*, or *CCTx*, the disable transition has no effect (command is ignored) and the message buffer state is not changed. No notification is given to the application.

The lock and unlock commands issued by writing 1 to the trigger bit MBCCSRn.LCKT. The transition that triggered by each of these commands depends on the current value of the status bit MBCCSRn.LCKS. If the command triggers the lock transition *HL* and the message buffer is in the state *CCTx,* the lock transition has no effect (command is ignored) and message buffer state is not changed. In this case, the message buffer lock error flag LCK_EF in the CHI Error Flag Register (CHIERFR) is set.

Table 3-92. Single Transmit Message Buffer Application Transitions

Transition	Command	Condition	Description
HE	MBCCSRn.EDT:= 1	MBCCSRn.EDS = 0	Application triggers message buffer enable.
HD		MBCCSRn.EDS = 1	Application triggers message buffer disable.

Table 3-92. Single Transmit Message Buffer Application Transitions

Transition	Command	Condition	Description
HL	MBCCSRn.LCKT:= 1	MBCCSRn.LCKS = 0	Application triggers message buffer lock.
HU		MBCCSRn.LCKS = 1	Application triggers message buffer unlock.

Module Transitions

The module transitions that can be triggered by the FlexRay module are described in Table 3-93. Each transition is triggered for certain message buffers when the related condition is fulfilled.

Table 3-93. Single Transmit Message Buffer Module Transitions

Transition	Condition	Description
SA	slot match and static slot	**Slot Assigned** - Message buffer is assigned to next static slot.
MA	slot match and CycleCounter match	**Message Available** - Message buffer is assigned to next slot and cycle counter filter matches.
TX	slot start and MBCCSRn.CMT = 1	**Transmission Slot Start** - Slot Start and commit bit CMT is set. In case of a dynamic slot, pLatestTx is not exceeded.
SU	status updated	**Status Updated** - Slot Status field and message buffer status flags updated. Interrupt flag set.
STS	static slot start	**Static Slot Start** - Start of static slot.
DSS	dynamic slot start or symbol window start or NIT start	**Dynamic Slot or Segment Start.** - Start of dynamic slot or symbol window or NIT.
SSS	slot start or symbol window start or NIT start	**Slot or Segment Start** - Start of static slot or dynamic slot or symbol window or NIT.

Transition Priorities

The application can trigger only one transition at a time. There is no need to specify priorities among them.

As shown in the first part of Table 3-94, the module transitions have a higher priority than the application transitions. For all states except the *CCMa* state, both a lock/unlock transition *HL/HD* and a module transition can be executed at the same time. The result state is reached by first applying the application transition and subsequently the module transition to the intermediately reached state. For example, if the message buffer is in the *HLck* state and the application unlocks the message buffer by the *HU* transition and the module triggers the slot assigned transition *SA*, the intermediate state is *Idle* and the resulting state is *CCSa*.

The priorities among the module transitions is given in the second part of Table 3-94.

Table 3-94. Single Transmit Message Buffer Transition Priorities

State	Priorities	Description
module vs. application		
Idle, HLck	SA > HD MA > HD	Slot Assigned > Message Buffer Disable Message Available > Message Buffer Disable
CCMa	TX > HL	Transmission Start > Message Buffer Lock

Table 3-94. Single Transmit Message Buffer Transition Priorities

State	Priorities	Description
module internal		
Idle, HLck	MA > SA	Message Available > Slot Assigned
CCMa	TX > STS TX > DSS	Transmission Slot Start > Static Slot Start Transmission Slot Start > Dynamic Slot Start

3.4.6.2.4 Transmit Message Setup

To transmit a message over the FlexRay bus, the application writes the message data into the message buffer data field and sets the commit bit CMT in the Message Buffer Configuration, Control, Status Registers (MBCCSRn). The physical access to the message buffer data field is described in Section 3.4.3.1, “Individual Message Buffers”.

As indicated by Table 3-91, the application shall write to the message buffer data field and change the commit bit CMT only if the transmit message buffer is in one of the states *HDis, HDisLck, HLck, HLckCCSa, HLckCCMa, or HLckCCMa.* The application can change the state of a message buffer if it issues the appropriate commands shown in Table 3-92. The state change is indicated through the MBCCSRn.EDS and MBCCSRn.LCKS status bits.

If the transmit message buffer enters one of the states *HDis, HDisLck, HLck, HLckCCSa, HLckCCMa, or HLckCCMa* the MBCCSRn.DVAL flag is negated.

3.4.6.2.5 Message Transmission

As a result of the message buffer search described in Section 3.4.7, “Individual Message Buffer Search”, the FlexRay module triggers the message available transition *MA* for up to two transmit message buffers. This changes the message buffer state from *Idle* to *CCMa* and the message buffers can be used for message transmission in the next slot.

The FlexRay module transmits a message from a message buffer if both of the following two conditions are fulfilled at the start of the transmission slot:

- The message buffer is in the message available state*CCMa*
- The message data remainsvalid, MBCCSRn.CMT equals 1

In this case, the FlexRay module triggers the *TX* transition and changes the message buffer state to *CCTx*. A transmit message buffer timing and state change diagram for message transmission is given in Figure 3-111. In this example, the message buffer with message buffer number n is *Idle* at the start of the search slot, matches the slot and cycle number of the next slot, and message buffer data are valid, MBCCSRn.CMT equals 1.

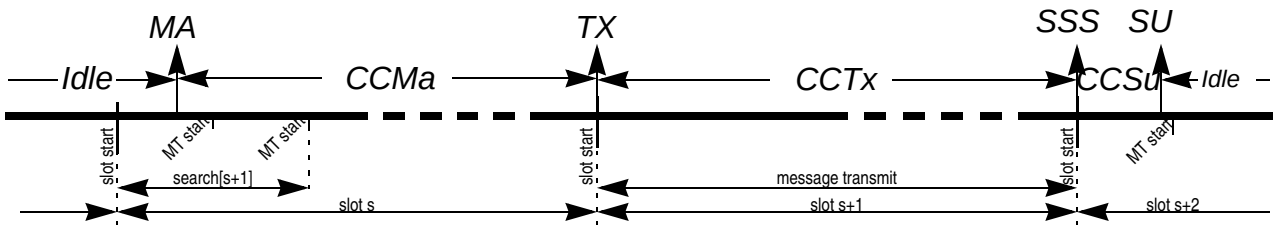


Figure 3-111. Message Transmission Timing

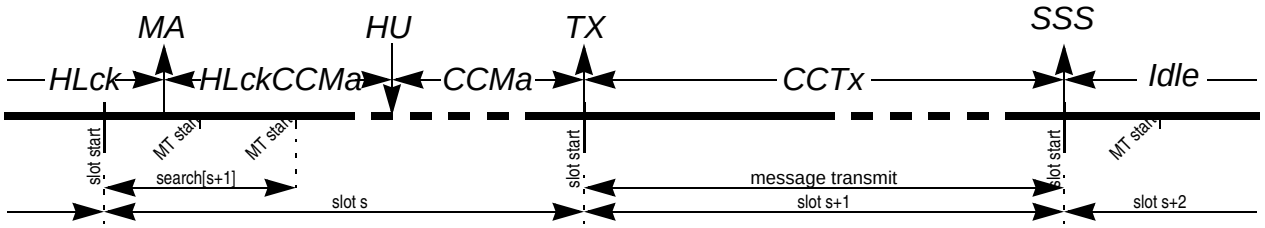


Figure 3-112. Message Transmission from HLck state with unlock

The amount of message data read from the FRM and transferred to the FlexRay bus is determined by the following three items

1. the message buffer segment that the message buffer is assigned to, as defined by the Message Buffer Segment Size and Utilization Register (MBSSUTR).
2. the message buffer data field size, as defined by the related field of the Message Buffer Data Size Register (MBDSR)
3. the value of the PLDLEN field in the message buffer header field, as described in Section 3.4.5.2.1, “Frame Header Section Description”

If a message buffer is assigned to message buffer segment 1, and PLDLEN is greater than MBSEG1DS, 2 * MBSEG1DS bytes are read from the message buffer data field and zero padding is used for the remaining bytes for the FlexRay bus transfer. If PLDLEN is less than or equals MBSEG1DS, the FlexRay module reads and transfers 2*PLDLEN bytes. The same holds for segment 2 and MBSEG2DS.

3.4.6.2.6 Null Frame Transmission

A static slot with slot number S is assigned to the FlexRay module for channel A, if at least one transmit message buffer is configured with the MBFIDRn.FID set to S and MBCCFRn.CHA set to 1. A Null Frame is transmitted in the static slot S on channel A if this slot is assigned to the FlexRay module for channel A, and all transmit message buffers with MBFIDRn.FID equaling s and MBCCFRn.CHA equaling 1 are not committed 9MBCCSRn.CMT = 0), locked by the application (MBCCSRn.LCKS = 1), or the cycle counter filter is enabled and does not match.

Additionally, the application can clear the commit bit of a message buffer that is in the *CCMa* state, which is called *uncommit* or *transmit abort*. This message buffer is used for null frame transmission.

As a result of the message buffer search described in Section 3.4.7, “Individual Message Buffer Search”, the FlexRay module triggers the slot assigned transition *SA* for up to two transmit message buffers if at

least one of the conditions mentioned above is fulfilled for these message buffers. The transition *SA* changes the message buffer states from *Idle* to *CCSa* or from *HLck* to *HLckCCSa*. In each case, these message buffers are used for null frame transmission in the next slot. A message buffer timing and state change diagram for null frame transmission from *Idle* state is given in Figure 3-113.

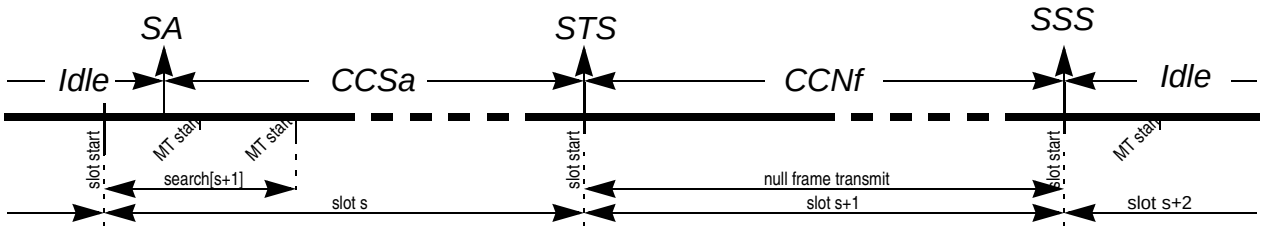


Figure 3-113. Null Frame Transmission from Idle state

A message buffer timing and state change diagram for null frame transmission from *HLck* state is given in Figure 3-114.

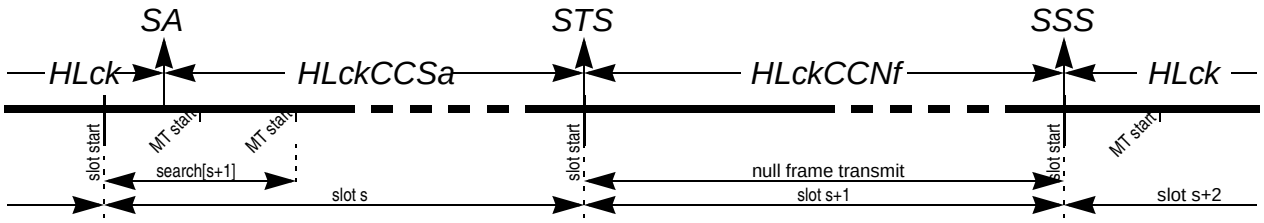


Figure 3-114. Null Frame Transmission from HLck state

If a transmit message buffer is in the *CCSa* or *HLckCCSa* state at the start of the transmission slot, a null frame is transmitted in any case, even if the message buffer is unlocked or committed before the transmission slot starts. A transmit message buffer timing and state change diagram for null frame transmission for this case is given in Figure 3-115.

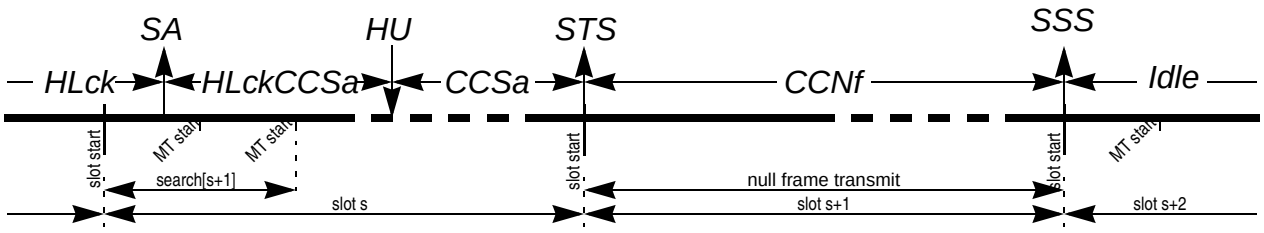


Figure 3-115. Null Frame Transmission from HLck state with unlock

Because the null frame transmission does not use the message buffer data, the application can lock/unlock the message buffer during null frame transmission. A transmit message buffer timing and state change diagram for null frame transmission for this case is given in Figure 3-116.

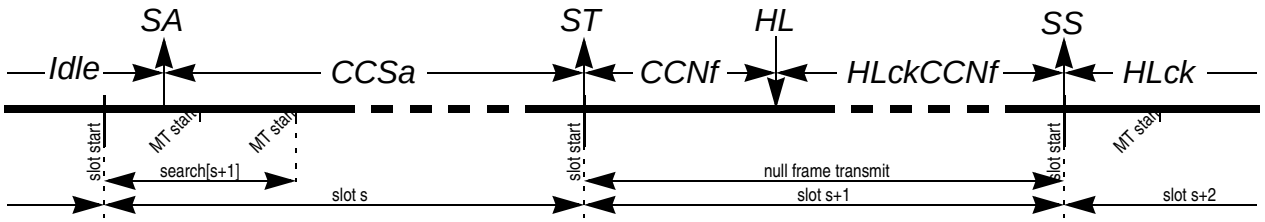


Figure 3-116. Null Frame Transmission from Idle State with locking

3.4.6.2.7 Message Buffer Status Update

After the end of each slot, the PE generates the slot status vector. Depending on the this status, the transmitted frame type, and the amount of transmitted data, the message buffer status is updated.

Message Buffer Status Update after Complete Message Transmission

The term complete message transmission refers to the fact that all payload data stored in the message buffer were send to FlexRay bus. In this case, the FlexRay module updates the slot status field of the message buffer and triggers the status updated transition *SU*. With the *SU* transition, the FlexRay module sets the message buffer interrupt flag MBCCSn.MBIF to indicate the successful message transmission.

Depending on the transmission mode flag MBCCFRn.MTM, the FlexRay module changes the commit flag MBCCSRn.CMT and the valid flag MBCCSRn.DVAL. If the MBCCFRn.MTM flag is negated, the message buffer is in the *event transmission mode*. In this case, each committed message is transmitted only once. The commit flag MBCCSRn.CMT is cleared with the *SU* transition. If the MBCCFRn.MTM flag is asserted, the message buffer is in the *state transmission mode*. In this case, each committed message is transmitted as long as the application provides new data or locks the message buffers. The FlexRay module does not clear the MBCCSRn.CMT flag at the end of transmission and sets the valid flag MBCCSRn.DVAL to indicate the message is transmitted again.

Message Buffer Status Update after Incomplete Message Transmission

The term incomplete message transmission refers to the fact that not all payload data that should be transmitted were send to FlexRay bus. This may be caused by the following regular conditions in the dynamic segment:

1. The transmission slot starts in a minislot with a minislot number greater than *pLatestTx*.
2. The transmission slot did not exist in the dynamic segment at all.

Additionally, an incomplete message transmission can be caused by internal communication errors. If those error occur, the Protocol Engine Communication Failure Interrupt Flag PECF_IF is set in the Protocol Interrupt Flag Register 1 (PIFR1).

In any of these two cases, the status of the message buffer is not changed at all with the SU transition. The slot status field is not updated, the status and control flags are not changed, and the interrupt flag is not set.

Message Buffer Status Update after Null Frame Transmission

After the transmission of a null frame, the status of the message buffer that was used for the null frame transmission is not changed at all. The slot status field is not updated, the status and control flags are not changed, and the interrupt flag is not set.

3.4.6.3 Receive Message Buffers

The section provides a detailed description of the functionality of the receive message buffers.

A receive message buffer is used to receive a message from the FlexRay Bus based on individual filter criteria. The FlexRay module uses the receive message buffer to provide the following data to the application

1. message data received
2. information about the reception process
3. status information about the slot in which the message was received

A individual message buffer with message buffer number *n* is configured as a receive message buffer by the following configuration settings

- MBCCSRn.MBT = 0 (single buffered message buffer)
- MBCCSRn.MTD = 0 (receive message buffer)

To certain message buffer fields, both the application and the FlexRay module have access. To ensure data consistency, a message buffer locking scheme is implemented that is used to control the access to the data, control, and status bits of a message buffer. The access regions for receive message buffers are depicted in Figure 3-117. A description of the regions is given in Table 3-95. If an region is active as indicated in Table 3-96, the access scheme given for that region applies to the message buffer.

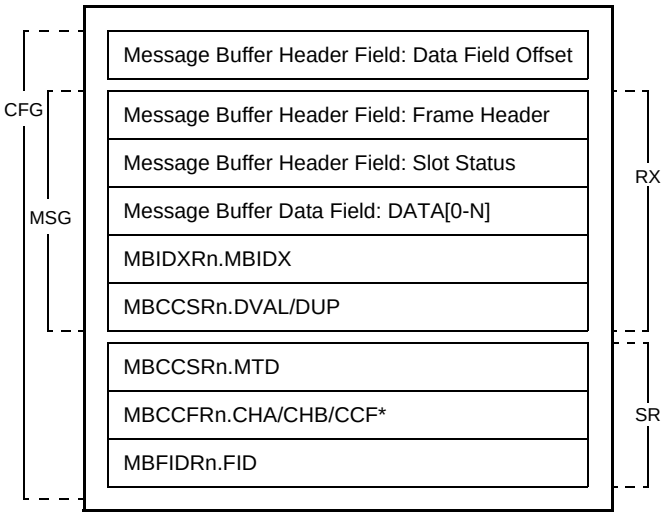


Figure 3-117. Receive Message Buffer Access Regions

Table 3-95. Receive Message Buffer Access Region Description

Region	Access from		Region used for
	Application	Module	
CFG	read/write	-	Message Buffer Configuration, Message Data and Status Access
MSG	read/write	-	Message Data, Header, and Status Access
RX	-	write-only	Message Reception and Status Update
SR	-	read-only	Message Buffer Search Data

The trigger bits MBCCSRn.EDT and MBCCSRn.LCKT and the interrupt enable bit MBCCSRn.MBIE are not under access control and can be accessed from the application at any time. The status bits MBCCSRn.EDS and MBCCSRn.LCKS are not under access control and can be accessed from the FlexRay module at any time.

The interrupt flag MBCCSRn.MBIF is not under access control and can be accessed from the application and the FlexRay module at any time. FlexRay module set access has higher priority.

The FlexRay module restricts its access to the regions depending on the current state of the message buffer. The application must adhere to these restrictions to ensure data consistency. The receive message buffer states are given in Figure 3-118. A description of the message buffer states is given in Table 3-91, which also provides the access scheme for the access regions.

The status bits MBCCSRn.EDS and MBCCSRn.LCKS provide the application with the required status information. The internal status information is not visible to the application.

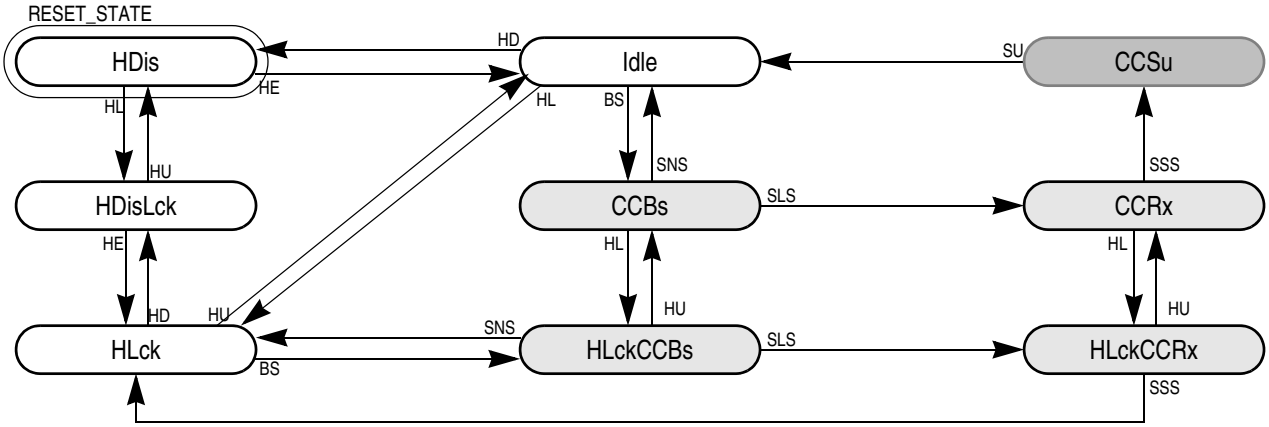


Figure 3-118. Receive Message Buffer States

Table 3-96. Receive Message Buffer States and Access

State	MBCCSRn		Access from		Description
	EDS	LCKS	Appl.	Module	
Idle	1	0	–	SR	**Idle** - Message Buffer is idle. Included in message buffer search.
HDis	0	0	CFG	–	**Disabled** - Message Buffer under configuration. Excluded from message buffer search.

Table 3-96. Receive Message Buffer States and Access (Continued)

State	MBCCSRn		Access from		Description
	EDS	LCKS	Appl.	Module	
HDisLck	0	1	CFG	–	**Disabled and Locked** - Message Buffer under configuration. Excluded from message buffer search.
HLck	1	1	MSG	–	**Locked** - Applications access to data, control, and status. Included in message buffer search.
CCBs	1	0	–	–	**Buffer Subscribed** - Message buffer subscribed for reception. Filter matches next (slot, cycle, channel) tuple.
HLckCCBs	1	1	MSG	–	**Locked and Buffer Subscribed** - Applications access to data, control, and status. Message buffer subscribed for reception.
CCRx	1	0	–	–	**Message Receive** - Message data received into related shadow buffer.
HLckCCRx	1	1	MSG	–	**Locked and Message Receive** - Applications access to data, control, and status. Message data received into related shadow buffer.
CCSu	1	0	–	RX	**Status Update** - Message buffer status update. Update of status flags, the slot status field, and the header index.

3.4.6.3.1 Message Buffer Transitions

Application Transitions

The application transitions that can be triggered by the application using the commands described in Table 3-97. The application issues the commands by writing to the Message Buffer Configuration, Control, Status Registers (MBCCSRn). Only one command can be issued with one write access. Each command is executed immediately. If the command is ignored, it must be issued again.

The enable and disable commands issued by writing 1 to the trigger bit MBCCSRn.EDT. The transition triggered by each of these command depends on the current value of the status bit MBCCSRn.EDS. If the command triggers the disable transition *HD* and the message buffer is in one of the states *CCBs*, *HLckCCBs*, or *CCRx*, the disable transition has no effect (command is ignored) and the message buffer state is not changed. No notification is given to the application.

The lock and unlock commands issued by writing 1 to the trigger bit MBCCSRn.LCKT. The transition triggered by each of these commands depends on the current value of the status bit MBCCSRn.LCKS. If the command triggers the lock transition *HL* while the message buffer is in the *state CCRx,* the lock transition has no effect (command is ignored) and message buffer state is not changed. In this case, the message buffer lock error flag LCK_EF in the CHI Error Flag Register (CHIERFR) is set.

Table 3-97. Receive Message Buffer Application Transitions

Transition	Host Command	Condition	Description
HE	MBCCSRn.EDT:= 1	MBCCSRn.EDS = 0	Application triggers message buffer enable.
HD		MBCCSRn.EDS = 1	Application triggers message buffer disable.
HL	MBCCSRn.LCKT:= 1	MBCCSRn.LCKS = 0	Application triggers message buffer lock.
HU		MBCCSRn.LCKS = 1	Application triggers message buffer unlock.

Module Transitions

The module transitions that can be triggered by the FlexRay module are described in Table 3-98. Each transition is triggered for certain message buffers when the related condition is fulfilled.

Table 3-98. Receive Message Buffer Module Transitions

Transition	Condition	Description
BS	slot match and CycleCounter match	**Buffer Subscribed** - The message buffer filter matches next slot and cycle.
SLS	slot start	**Slot Start** - Start of Static Slot or Dynamic Slot.
SNS	symbol window start or NIT start	**Symbol Window or NIT Start** - Start of Symbol Window or NIT.
SSS	slot start or symbol window start or NIT start	**Slot or Segment Start** - Start of Static Slot, Dynamic Slot, Symbol Window, or NIT.
SU	status updated	**Status Updated** - Slot Status field, message buffer status flags, header index updated. Interrupt flag set.

Transition Priorities

The application can trigger only one transition at a time. There is no need to specify priorities among them.

As shown in Table 3-99, the module transitions have a higher priority than the application transitions. For all states except the *CCRx* state, a module transition and the application lock/unlock transition *HL/HU* and can be executed at the same time. The result state is reached by first applying the module transition and subsequently the application transition to the intermediately reached state. For example, if the message buffer is in the buffer subscribed state *CCBs* and the module triggers the slot start transition *SLS* at the same time as the application locks the message buffer by the *HL* transition, the intermediate state is *CCRx* and the resulting state is locked buffer subscribed state *HLckCCRx*.

Table 3-99. Receive Message Buffer Transition Priorities

State	Priorities	Description
module vs. application		
Idle	BS > HD	Buffer Subscribed > Message Buffer Disable
HLck	BS > HD	Buffer Subscribed > Message Buffer Disable
CCRx	SSS > HL	Slot or Segment Start > Message Buffer Lock

3.4.6.3.2 Message Buffer Search

The FlexRay module starts a sequential search that checks all message buffers at the following protocol related events:

- slot start, in the static frame segment
- minislot start, in the dynamic frame segment
- NIT start

The filters that are used for the search are described in Section 3.4.7.1, “Individual Message Buffer Filtering”.

As a result of the message buffer search, the FlexRay module changes the state of up to two enabled receive message buffers from idle state *Idle* or locked state *HLck* to the subscribed state *CCBs* or locked buffer subscribed state *HLckCCBs* by triggering the buffer subscribed transition *BS.*

If the receive message buffers for the next slot are assigned to both channels, then at most one receive message buffer is changed to a buffer subscribed state.

If more than one matching message buffers assigned to a certain channel, then only the message buffer with the lowest message buffer number is in one of the states mentioned above.

3.4.6.3.3 Message Reception

With the start of the next static or dynamic slot the module trigger the slot start transition *SLS*. This changes the state of the subscribed receive message buffers from *CCBs* to *CCRx* or from *HLckCCBs* to *HLckCCRx,* respectively.

During the reception slot, the received frame data are written into the shadow buffers. For details on receive shadow buffers, see Section 3.4.6.3.6, “Receive Shadow Buffers Concept”. The data and status of the receive message buffers that are the *CCRx* or *HLckCCRx* are not modified in the reception slot.

3.4.6.3.4 Message Buffer Status Update

With the start of the next static or dynamic slot or with the start of the symbol window or NIT, the module trigger the slot or segment start transition *SSS*. This transition changes the state of the receiving receive message buffers from *CCRx* to *CCSu* or from *HLckCCRx* to *HLck,* respectively.

If a message buffer was in the locked state *HLckCCRx*, no update is performed. The received data are lost. This is indicated by setting the Frame Lost Channel A/B Error Flag FRLA_EF/FRLB_EF in the CHI Error Flag Register (CHIERFR).

If a message buffer was in the *CCRx* state it is now in the *CCSu* state. After the evaluation of the slot status provided by the PE the message buffer is updated. The message buffer update depends on the slot status bits and the segment the message buffer is assigned to. This is described in Table 3-100.

Table 3-100. Receive Message Buffer Update

vSS!ValidFrame	*vRF!Header!NFIndicator*	Update description
1	1	**Valid non-null frame received.** - Message Buffer Data Field updated. - Frame Header Field updated. - Slot Status Field updated. - DUP:= 1 - DVAL:= 1 - MBIF:= 1
1	0	**Valid null frame received.** - Message Buffer Data Field *not* updated. - Frame Header Field *not* updated. - Slot Status Field updated. - DUP:= 0 - DVAL *not* changed - MBIF:= 1

Table 3-100. Receive Message Buffer Update (Continued)

vSS!ValidFrame	*vRF!Header!NFIndicator*	Update description
0	x	**No valid frame received.** - Message Buffer Data Field not updated. - Frame Header Field not updated. - Slot Status Field updated. - DUP:= 0 - DVAL *not* changed. - MBIF:= 1, if the slot was not an empty dynamic slot. **Note:** An empty dynamic slot is indicated by the following frame and slot status bit values: *vSS!ValidFrame* = 0 and *vSS!SyntaxError* = 0 and *vSS!ContentError* = 0 and *vSS!BViolation* = 0.

NOTE

If the number of the last slot in the current communication cycle on a given channel is *n*, all receive message buffers assigned to this channel with MBFIDRn.FID greater than *n* are not updated at all.

When the receive message buffer update has finished the status updated transition *SU* is triggered, which changes the buffer state from *CCSu* to *Idle*. An example receive message buffer timing and state change diagram for a normal frame reception is given in Figure 3-119.

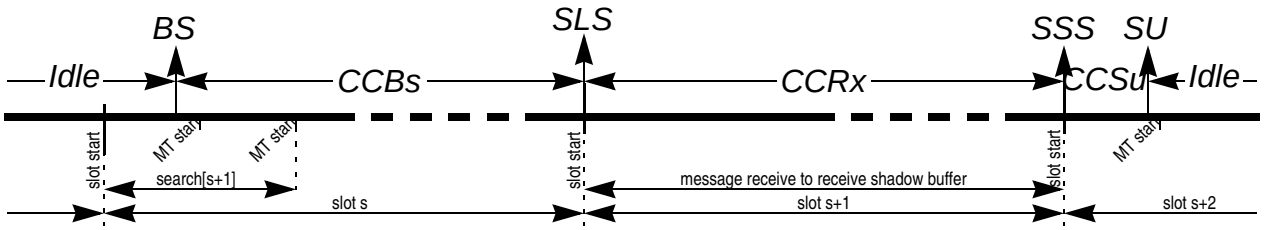


Figure 3-119. Message Reception Timing

The amount of message data written into the message buffer data field of the receive shadow buffer is determined by the following two items:

1. the message buffer segment that the message buffer is assigned to, as defined by the Message Buffer Segment Size and Utilization Register (MBSSUTR).
2. the message buffer data field size, as defined by the related field of the Message Buffer Data Size Register (MBDSR)
3. the number of bytes received over the FlexRay bus

If the message buffer is assigned to the message buffer segment 1, and the number of received bytes is greater than 2*MBDSR.MBSEG1DS, the FlexRay module writes only 2*MBDSR.MBSEG1DS bytes into the message buffer data field of the receive shadow buffer. If the number of received bytes is less than 2*MBDSR.MBSEG1DS, the FlexRay module writes only the received number of bytes and does not change the trailing bytes in the message buffer data field of the receive shadow buffer. The same holds for the message buffer segment 2 with MBDSR.MBSEG2DS.

3.4.6.3.5 Received Message Access

To access the message data received over the FlexRay bus, the application reads the message data stored in the message buffer data field of the corresponding receive message buffer. The access to the message buffer data field is described in Section 3.4.3.1, “Individual Message Buffers”.

The application can read the message buffer data field if the receive message buffer is one of the states *HDis, HDisLck, or HLck.* If the message buffer is in one of these states, the FlexRay module does not change the content of the message buffer.

3.4.6.3.6 Receive Shadow Buffers Concept

The receive shadow buffer concept applies only to individual receive message buffers. The intention of this concept is to ensure that only syntactically and semantically valid received non-null frames are presented to the application in a receive message buffer. The basic structure of a receive shadow buffer is described in Section 3.4.3.2, “Receive Shadow Buffers”.

The receive shadow buffers temporarily store the received frame header and message data. After the slot boundary the slot status information is generated. If the slot status information indicates the reception of the valid non-null frame (see Table 3-100), the FlexRay module writes the slot status into the slot status field of the receive shadow buffer and exchanges the content of the Message Buffer Index Registers (MBIDXRn) with the content of the corresponding internal shadow buffer index register. In all other cases, the FlexRay module writes the slot status into the identified receive message buffer, depending on the slot status and the FlexRay segment the message buffer is assigned to.

The shadow buffer concept, with its index exchange, results in the fact that the FRM located message buffer associated to an individual receive message buffer changes after successful reception of a valid frame. This means that the message buffer area in the FRM accessed by the application for reading the received message is different from the initial setting of the message buffer. Therefore, the application must not rely on the index information written initially into the Message Buffer Index Registers (MBIDXRn). Instead, the index of the message buffer header field must be fetched from the Message Buffer Index Registers (MBIDXRn).

3.4.6.4 Double Transmit Message Buffer

The section provides a detailed description of the functionality of the double transmit message buffers.

Double transmit message buffers are used by the application to provide the FlexRay module with the message data to be transmitted over the FlexRay Bus. The FlexRay module uses this message buffer to provide information to the application about the transmission process, and status information about the slot in which message data was transmitted.

In contrast to the single transmit message buffers, the application can provide new transmission data while the transmission of the previously provided message data is running. This scheme is called double buffering and can be considered as a FIFO of depth 2.

Double transmit message buffers are implemented by combining two individual message buffers that form the two sides of an double transmit message buffer. One side is called the commit side and is accessed by the application to provide the message data. The other side is called the transmit side and is used by the

FlexRay module to transmit the message data to the FlexRay bus. The two sides are located in adjacent individual message buffers. The message buffer that implements the commit side has an even message buffer number 2n. The transmit side message buffer follows the commit side message buffer and has the message buffer number 2n+1. The basic structure and data flow of a double transmit message buffer is given in Figure 3-120.

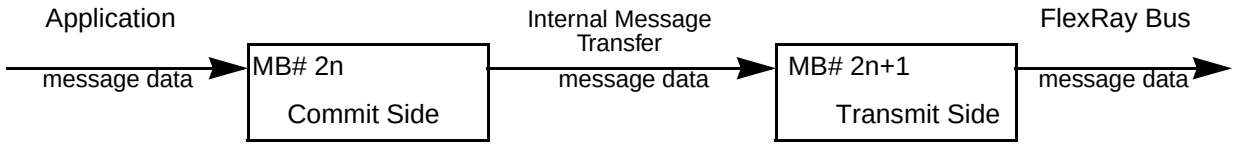


Figure 3-120. Double Transmit Buffer Structure and Data Flow

NOTE

Both the commit and the transmit side must be configured with identical values except for the Message Buffer Index Registers (MBIDXRn).

3.4.6.4.1 Access Regions

To certain message buffer fields, both the application and the FlexRay module have access. To ensure data consistency, a message buffer locking scheme is implemented, which controls the exclusive access to the data, control, and status bits of the message buffer.

The access scheme for double transmit message buffers is depicted in Figure 3-121. The given regions represent fields that can be accessed from both the application and the FlexRay module and, thus, require access restrictions. A description of the regions is given in Table 3-101.

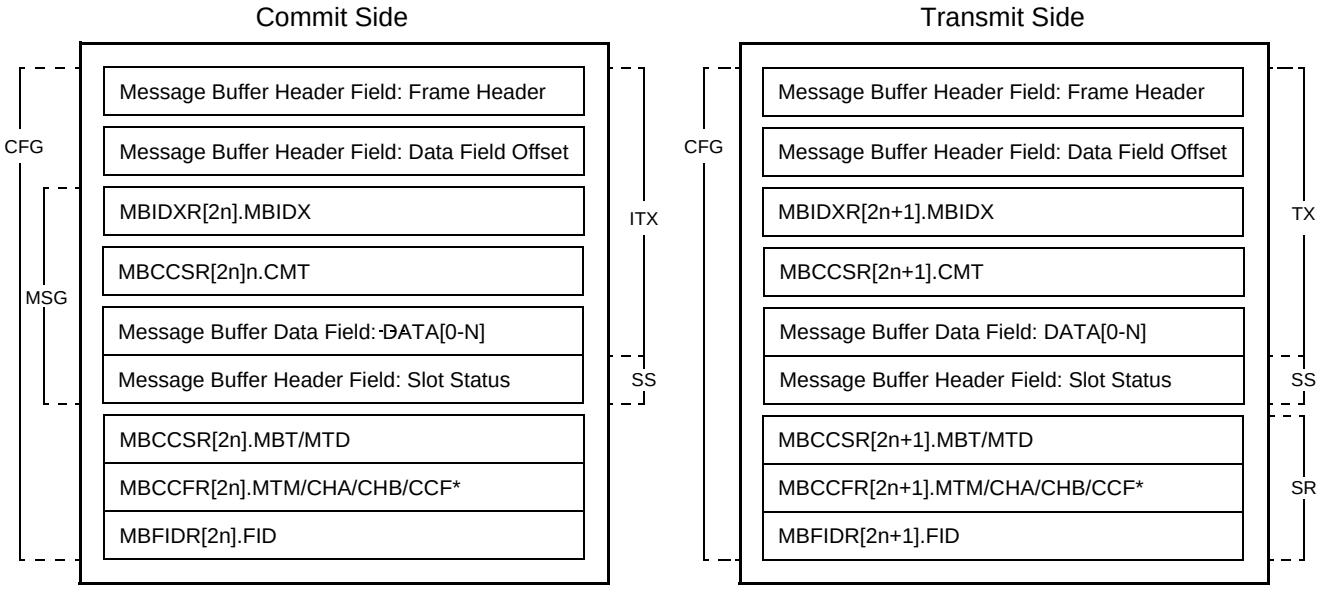


Figure 3-121. Double Transmit Message Buffer Access Regions Layout

Table 3-101. Double Transmit Message Buffer Access Regions Description

Access			Description
Region	Type		
	Application	Module	
Commit Side			
CFG	read/write	-	Message Buffer Configuration
MSG	read/write	-	Message Buffer Data and Control access
ITX	-	read/write	Internal Message Transfer.
SS	-	write-only	Slot Status Update
Transmit Side			
CFG	read/write	-	Message Buffer Configuration
SR	-	read-only	Message Buffer Search
TX	-	read-only	Internal Message Transfer, Message Transmission
SS	-	write-only	Slot Status Update

The trigger bits MBCCSRn.EDT and MBCCSRn.LCKT, and the interrupt enable bit MBCCSRn.MBIE are not under access control and can be accessed from the application at any time. The status bits MBCCSRn.EDS and MBCCSRn.LCKS are not under access control and can be accessed from the FlexRay module at any time.

The interrupt flag MBCCSnR.MBIF is not under access control and can be accessed from the application and the FlexRay module at any time. FlexRay module set access has higher priority.

The FlexRay module restricts its access to the regions, depending on the current state of the corresponding part of the double transmit message buffer. The application must adhere to these restrictions to ensure data consistency. The states for the commit side of a double transmit message buffer are given in Figure 3-122. A description of the states is given in Table 3-103. The states for the transmit side of a double transmit message buffer are given in Figure 3-123. A description of the states is given in Table 3-103. The description tables also provide the access scheme for the access regions.

The status bits MBCCSRn.EDS and MBCCSRn.LCKS provide the application with the required message buffer status information. The internal status information is not visible to the application.

3.4.6.4.2 Message Buffer States

This section describes the transmit message buffer states and provides a state diagram.

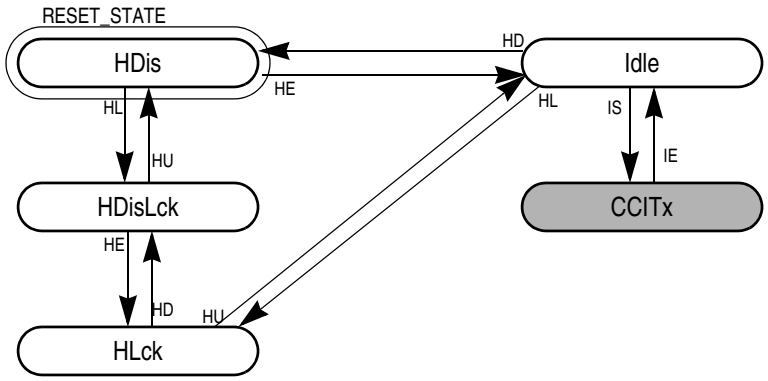


Figure 3-122. Double Transmit Message Buffer State Diagram (Commit Side)

A description of the states of the commit side of a double transmit message buffer is given in Table 3-102.

Table 3-102. Double Transmit Message Buffer State Description (Commit Side)

State	MBCCSR[2n]		Access Region		Description
	EDS	LCKS	Appl.	Module	
common states					
HDis	0	0	CFG	–	**Disabled** - Message Buffer under configuration. Commit Side can *not* be used for internal message transfer.
CCITx	1	0	–	ITX	**Internal Message Transfer** - Message Buffer Data transferred from commit side to transmit side.
commit side specific states					
Idle	1	0	–	ITX, SS	**Idle** - Message Buffer Commit Side is idle. Commit Side can be used for internal message transfer.
HDisLck	0	1	CFG	SS	**Disabled and Locked** - Message Buffer under configuration. Commit Side can *not* be used for internal message transfer.
HLck	1	1	MSG	SS	**Locked** - Applications access to data, control, and status. Commit Side can *not* be used for internal message transfer.

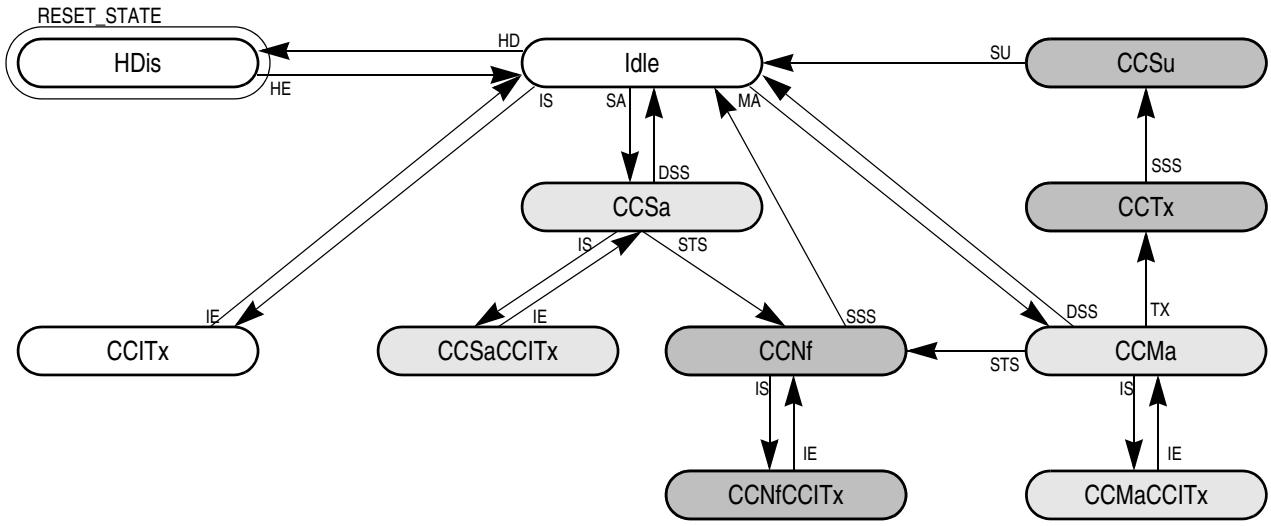


Figure 3-123. Double Transmit Message Buffer State Diagram (Transmit Side)

A description of the states of the transmit side of a double transmit message buffer is given in Table 3-103.

Table 3-103. Double Transmit Message Buffer State Description (Transmit Side)

State	MBCCSRn		Access Region		Description
	EDS	LCKS	Appl.	Module	
common states					
HDis	0	0	CFG	–	**Disabled** - Message Buffer under configuration. Excluded from message buffer search.
CCITx	1	0	–	TX	**Internal Message Transfer** - Message Buffer Data transferred from commit side to transmit side.
transmit side specific states					
Idle	1	0	–	SR	**Idle** - Message Buffer Transmit Side is idle. Transmit Side is included in message buffer search.
CCSa	1	0	–	–	**Slot Assigned** - Message buffer assigned to next static slot. Ready for Null Frame transmission.
CCSaCCITx	1	0	–	TX	**Slot Assigned and Internal Message Transfer** - Message buffer assigned to next static slot and Message Buffer Data transferred from commit side to transmit side.
CCNf	1	0	–	TX	**Null Frame Transmission** Header is used for null frame transmission.
CCNfCCITx	1	0	–	TX	**Null Frame Transmission and Internal Message Transfer** - Header is used for null frame transmission and Message Buffer Data transferred from commit side to transmit side.
CCMa	1	0	–	–	**Message Available** - Message buffer is assigned to next slot and cycle counter filter matches.
CCMaCCITx	1	0	–	–	**Message Available and Internal Message Transfer** - Message buffer is assigned to next slot and cycle counter filter matches and Message Buffer Data transferred from commit side to transmit side.
CCTx	1	0	–	TX	**Message Transmission** - Message buffer data transmit. Payload data from buffer transmitted

Table 3-103. Double Transmit Message Buffer State Description (Transmit Side) (Continued)

State	MBCCSRn		Access Region		Description
	EDS	LCKS	Appl.	Module	
CCSu	1	0	–	SS	Status Update. Message buffer status update. Update of status flags, the slot status field, and the header index. **Note:** The slot status field of the commit side is updated too, even if the application has locked the commit side.

3.4.6.4.3 Message Buffer Transitions

Application Transitions

The application transitions that can be triggered by the application using the commands described in Table 3-104. The application issues the commands by writing to the Message Buffer Configuration, Control, Status Registers (MBCCSRn). Only one command can be issued with one write access. Each command is executed immediately. If the command is ignored, it must be issued again.

The enable and disable commands can be issued on the transmit side only. Any enable or disable command issued on the commit side is ignored without notification. The transitions triggered depend on the value of the EDS bit. The enable and disable commands affect the commit side and the transmit side at the same time. If the application triggers the disable transition *HD* while the transmit side is in one of the states *CCSa*, *CCSaCCITx*, *CCNf*, *CCNfCCITx*, *CCMa*, *CCMaCCITx, CCTx, or CCSu,* the disable transition has no effect (command is ignored) and the message buffer state is not changed. No notification is given to the application.

The lock and unlock commands can be issued on the commit side only. Any lock or unlock command issued on the transmit side are ignored and the double transmit buffer lock error flag DBL_EF in the CHI Error Flag Register (CHIERFR) is set. The transitions triggered depend on the current value of the LCKS bit. The lock and unlock commands only affect the commit side. If the application triggers the lock transition *HL* while the commit side is in the state *CCITx,* the message buffer state is not changed and the message buffer lock error flag LCK_EF in the CHI Error Flag Register (CHIERFR) is set.

Table 3-104. Double Transmit Message Buffer Host Transitions

Transition	Host Command	Condition	Description
HE	MBCCSR[2n+1].EDT:= 1	MBCCSR[2n+1].EDS = 0	Application triggers message buffer enable.
HD		MBCCSR[2n+1].EDS = 1	Application triggers message buffer disable.
HL	MBCCSR[2n].LCKT:= 1	MBCCSR[2n].LCKS = 0	Application triggers message buffer lock.
HU		MBCCSR[2n].LCKS = 1	Application triggers message buffer unlock.

Module Transitions

The module transitions that can be triggered by the FlexRay module are described in Table 3-105. The transitions *C1* and *C2* apply to both sides of the message buffer and are applied at the same time. All other FlexRay module transitions apply to the transmit side only.

Table 3-105. Double Transmit Message Buffer Module Transitions

Transition	Condition	Description
common transitions		
IS	see Section 3.4.6.4.5, "Internal Message Transfer	**Internal Message Transfer Start** - Start transfer of message data from commit side to transmit side.
IE		**Internal Message Transfer End** - Stop transfer of message data from commit side to transmit side. **Note:** The internal message transfer is stopped before the slot or segment start.
transmit side specific transitions		
SA	slot match and static slot	**Slot Assigned** - Message buffer is assigned to next static slot.
MA	slot match and CycleCounter match	**Message Available** - Message buffer is assigned to next slot and cycle counter filter matches.
TX	slot start and MBCCSR[2n+1].CMT = 1	**Transmission Slot Start** - Slot Start and commit bit CMT is set. In case of a dynamic slot, pLatestTx is not exceeded.
SU	status updated	**Status Updated** - Slot Status field and message buffer status flags updated. Interrupt flag set.
STS	static slot start	**Static Slot Start** - Start of static slot.
DSS	dynamic slot start or symbol window start or NIT start	**Dynamic Slot or Segment Start.** - Start of dynamic slot or symbol window or NIT.
SSS	slot start or symbol window start or NIT start	**Slot or Segment Start** - Start of static slot or dynamic slot or symbol window or NIT.

Transition Priorities

The application can trigger only one transition at a time. There is no need to specify priorities among them.

As shown in the first part of Table 3-106, the module transitions have a higher priority than the application transitions. The priorities among the FlexRay module transitions and the related states are given in the second part of Table 3-106. These priorities apply only to the transmit side. The internal message transmit start transition IS has tho lowest priority.

Table 3-106. Double Transmit Message Buffer Transition Priorities

State	Priority	Description
module vs. application		
Idle	IS > HD IS > HL	Internal Message Transfer Start > Message Buffer Disable Internal Message Transfer Start > Message Buffer Lock
module internal		
Idle	MA > SA	Message Available > Slot Assigned
CCMa	TX > STS TX > DSS	Transmission Slot Start > Static Slot Start Transmission Slot Start > Dynamic Slot Start

3.4.6.4.4 Message Preparation

The application provides the message data through the commit side. The transmission itself is executed from the transmit side. The transfer of the message data from the commit side to the transmit side is done by the *Internal Message Transfer*, which is described in Section 3.4.6.4.5, “Internal Message Transfer

To transmit a message over the FlexRay bus, the application writes the message data into the message buffer data field of the commit side and sets the commit bit CMT in the Message Buffer Configuration, Control, Status Registers (MBCCSRn). The physical access to the message buffer data field is described in Section 3.4.3.1, “Individual Message Buffers”.

As indicated by Table 3-102, the application shall write to the message buffer data field and change the commit bit CMT only if the transmit message buffer is in one of the states *HDis, HDisLck, or HLck*. The application can change the state of a message buffer if it issues the appropriate commands shown in Table 3-104. The state change is indicated through the MBCCSRn.EDS and MBCCSRn.LCKS status bits.

3.4.6.4.5 Internal Message Transfer

The internal message transfer transfers the message data from the commit side to the transmit side. The internal message transfer is implemented as the swapping of the content of the Message Buffer Index Registers (MBIDXRn) of the commit side and the transmit side. After the swapping, the commit side CMT bit is cleared, the commit side interrupt flag MBIF is set, the transmit side CMT bit is set, and the transmit side DVAL bit is cleared.

The conditions and the point in time when the internal message transfer is started are controlled by the message buffer commit mode bit MCM in the Message Buffer Configuration, Control, Status Registers (MBCCSRn). The MCM bit configures the message buffer for the streaming commit mode or the immediate commit mode. A detailed description is given in Streaming Commit Mode and Immediate Commit Mode. The Internal Message Transfer is triggered with the transition *IS*. Both sides of the message buffer enter enter one of the CC*ITx* states. The internal message transfer is finished with the transition *IE*.

Streaming Commit Mode

The intention of the streaming commit mode is to ensure that each committed message is transmitted at least once. The FlexRay module does not start the Internal Message Transfer for a message buffer as long as the message data on the transmit side is not transmitted at least once.

The streaming commit mode is configured by clearing the message buffer commit mode bit MCM in the Message Buffer Configuration, Control, Status Registers (MBCCSRn).

In this mode, the internal message transfer from the commit side to the transmit side is started for a double transmit message buffer when all of the following conditions are fulfilled:

- The commit side is in the idle state
- The commit site message data isvalid, MBCCSR[2n].CMT equals 1
- The transmit side is in one ofthe states idle, CCSa, or CCMa
- The transmit side contains novalid message data, MBCCSR[2n+1].CMT equals 0, or the message data was transmitted at least once, MBCCSR[2n+1].DVAL equals 1

An example of a streaming commit mode state change diagram is given in Figure 3-124. In this example, both the commit and the transmit side do not contain valid message data and the application provides two messages. The message buffer does not match the next slot.

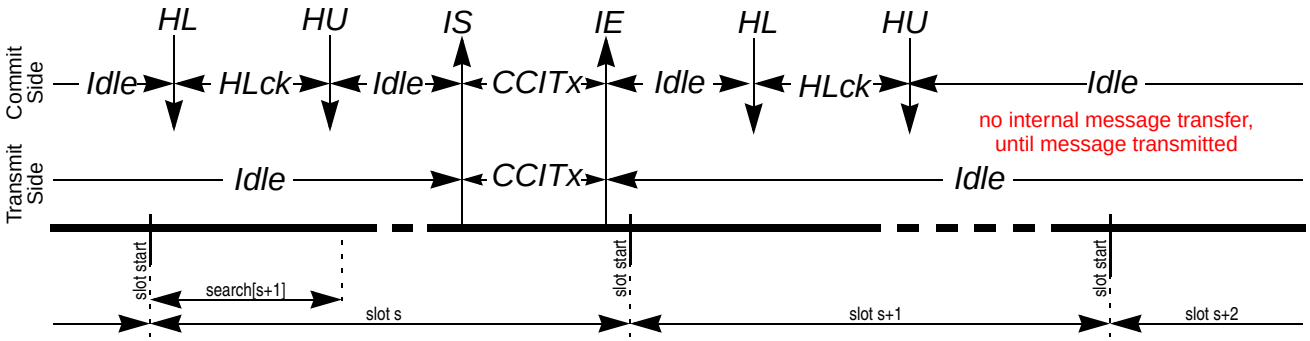


Figure 3-124. Internal Message Transfer in Streaming Commit Mode

Immediate Commit Mode

The intention of the immediate commit mode is to transmit the *latest* data provided by the application. This implies that it is not guaranteed that each provided message is transmitted at least once.

The immediate commit mode is configured by setting the message buffer commit mode bit MCM in the Message Buffer Configuration, Control, Status Registers (MBCCSRn).

In this mode, the internal message transfer from the commit side to the transmit side is started for one double transmit message buffer when all of the following conditions are fulfilled

1. the commit side is in the idle state
2. the commit site message data are valid, MBCCSR[2n].CMT equals 1
3. the transmit side is in one of the states idle, CCSa, or CCMa

It is not checked whether the transmit side contains no valid message data or valid message data were transmitted at least once. If message data are valid and not transmitted, they may be overwritten.

An example of a streaming commit mode state change diagram is given in Figure 3-125. In this example, both the commit and the transmit side do not contain valid message data, and the application provides two messages and the first message is gets overwritten. The message buffer does not match the next slot.

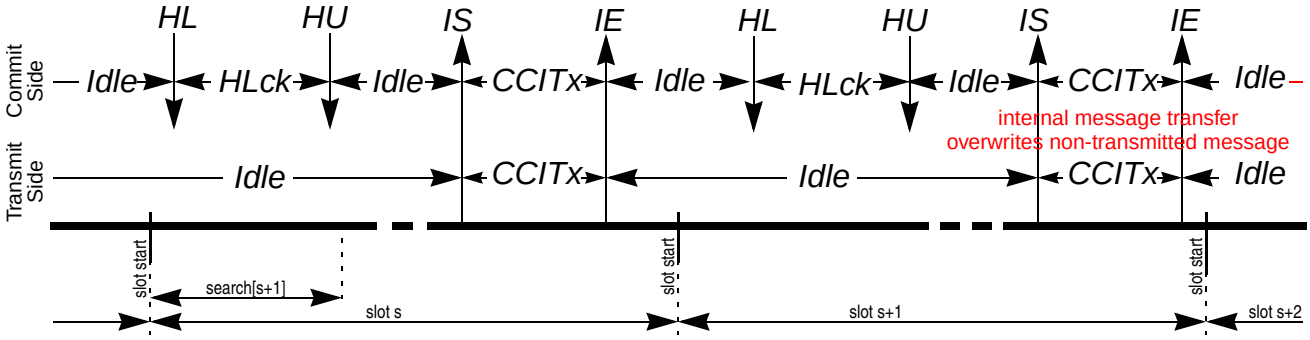


Figure 3-125. Internal Message Transfer in Immediate Commit Mode

3.4.6.4.6 Message Transmission

For double transmit message buffers, the message buffer search checks only the transmit side part. The internal scheduling ensures, that the internal message transfer is stopped on the message buffer search start. Thus, the transmit side of message buffer, that is not in its transmission or status update slot, is always in the *Idle* state.

The message transmit behavior and transmission state changes of the transmit side of a double transmit message buffer are the same as for single buffered transmit buffers, except that the transmit side of double buffers can not be locked by the application, i.e. the *HU* and *HL* transition do not exist. Therefore, refer to Section 3.4.6.2.5, “Message Transmission”

3.4.6.4.7 Message Buffer Status Update

The message buffer status update behavior of the transmit side of a double transmit message buffer is the same as for single transmit message buffers which is described in Section 3.4.6.2.7, “Message Buffer Status Update”.

Additionally, the slot status field of the commit side is update after the update of the slot status field of the transmit side, even if the commit side is locked by the application. This is implemented to provide the slot status of the most recent transmission slot.

3.4.7 Individual Message Buffer Search

This section provides a detailed description of the message buffer search algorithm.

The message buffer search checks all enabled individual message buffer to determine if a certain slot is assigned to this node for transmission or if this node is subscribed to a certain slot for reception. The message buffer search is a sequential algorithm and is started at the following protocol related events:

- each NIT start
- each slot start in the static frame segment
- each minislot start in the dynamic frame segment

The search within the NIT searches for message buffers assigned or subscribed to slot 1. The search within slot *n* searches for message buffers assigned or subscribed to slot *n+1*.

If the message buffer search is running while the next message buffer search start event appears, the message buffer search is stopped and the Message Buffer Search Error Flag MSB_EF is set in the CHI Error Flag Register (CHIERFR). This appears only if the CHI frequency is to low to search through all message buffers within the NIT or a minislot. The message buffer result is not defined in this case. For more details see Section 3.6.2, “Number of Usable Message Buffers”.

The filters criteria used for the message buffer search described in Section 3.4.7.1, “Individual Message Buffer Filtering”. For double transmit message buffers only the transmit side is included in the search. During the search, a list of all matching message buffers is created. If all message buffers assigned or subscribed to the next slot are assigned to only one channel, two lists of matching message buffers are created, one for each channel. If all message buffers assigned or subscribed to the next slot are assigned to both channels, only one sorted list of matching message buffers is created.

Each message buffer list is sorted according to the priorities given in Table 3-107. From the group with the highest priority, the message buffer with the lowest message buffer number is selected. For this message buffer the corresponding transition given in Table 3-107 is triggered as the result of the message buffer search.

Table 3-107. Message Buffer Search Priority

Priority	MBCCSRn				Description	Transition
	MTD	LCKS	CMT	CCFM[1]		
(highest) 0	1	0	1	1	transmit buffer, unlocked, committed, matches cycle count	MA
1	1	-	0	1	transmit buffer, *uncommitted,* matches cycle count	SA
	1	1	-	1	transmit buffer, locked, matches cycle count	SA
2	1	-	-	-	transmit buffer, assigned to slot	SA
3	0	0	n/a	1	*receive* buffer, unlocked, matches cycle count	SB
(lowest) 4	0	1	n/a	1	receive buffer, *locked,* matches cycle count	SB

[1] Cycle Counter Filter Match, see Section 3.4.7.1.2, “Message Buffer Cycle Counter Filtering”

3.4.7.1 Individual Message Buffer Filtering

The message buffer search identifies the matching message buffers by applying two individual message buffer filter. The first filter is the frame ID filter, the second filter is the cycle count filter.

3.4.7.1.1 Message Buffer Frame ID Filtering

The message buffer frame ID filter is used to determine if the message buffer can be considered for reception or transmission in a certain slot on a per channel basis.

The frame ID filter matches for a message buffer with message buffer number *n* and the search slot *s*, if the value of the FID field in the Message Buffer Frame ID Registers (MBFIDRn) equals *s*.

Only message buffer with a frame ID filter match may appear in the matching message buffer list. All transmit message buffer with a matching frame ID appear in the matching message buffer list. Only receive message buffer with a matching frame ID and a matching cycle counter filter appear in the matching message buffer list.

3.4.7.1.2 Message Buffer Cycle Counter Filtering

The message buffer cycle counter filter is a value-mask filter defined by the CCFE, CCFMSK, and CCFVAL fields in the Message Buffer Cycle Counter Filter Registers (MBCCFRn). This filter determines a set of communication cycles in which the message buffer is considered for message reception or message transmission. If the cycle counter filter is disabled, i.e. CCFE equals 0, this set of cycles consists of all communication cycles.

If the cycle counter filter of a message buffer does not match a certain communication cycle number, this message buffer is not considered for message transmission or reception in that communication cycle. In case of a transmit message buffer, though, this buffer is added to the matching message buffer list with

CCFM equaling 0 to indicate the slot assignment and to trigger the null frame transmission. In case of an receive message buffer, this buffer is *not* added to the matching message buffer list.

A message buffer matches its cycle counter filter for the communication cycle with the number CYCCNT if at least one of the following conditions evaluates to true:

$$\text{MBCCFRn[CCFE]} = 0 \qquad \textit{Eqn. 3-8}$$

$$\text{CYCCNT} \wedge \text{MBCCFRn[CCFMSK]} = \text{MBCCFRn[CCFVAL]} \wedge \text{MBCCFRn[CCFMSK]} \qquad \textit{Eqn. 3-9}$$

3.4.7.1.3 Message Buffer Channel Assignment Consistency

The message buffer channel assignment given by the CHA and CHB bits in the Message Buffer Cycle Counter Filter Registers (MBCCFRn) defines the channels on which the message buffer receive or transmit. The message buffer with number *n* transmits or receives on channel A if MBCCFRn.CHA equals 1 and transmits or receives on channel B if MBCCFRn.CHB equals 1.

To ensure correct message buffer operation, all message buffers assigned to the same slot must have a consistent channel assignment. That means that all message buffers assigned to the same slot must be assigned to only one channel, or assigned to both channels. The behavior of the message buffer search is not defined, if both types of channel assignments occur for one slot. An inconsistent channel assignment for message buffer 0 and message buffer 1 is depicted in Figure 3-126.

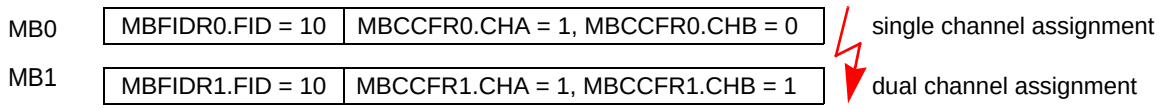


Figure 3-126. Inconsistent Channel Assignment

3.4.8 Individual Message Buffer Reconfiguration

The initial configuration of each individual message buffer can be changed even when the protocol is not in the *POC:config* state. This is referred to as individual message buffer *reconfiguration*. The configuration bits and fields that can be changed are given in the section on Specific Configuration Data. The common configuration data given in the section on Specific Configuration Data can not be reconfigured when the protocol is out of the *POC:config* state.

3.4.8.1 Reconfiguration Schemes

Depending on the target and destination basic state of the message buffer that is to be reconfigured, there are three reconfiguration schemes.

3.4.8.1.1 Basic Type Not Changed (RC1)

A reconfiguration does not change the basic type of the individual message buffer, if both the message buffer transfer direction bit MBCCSn.MTD and the message buffer type bit MBCCSn.MBT are not changed. This type of reconfiguration is denoted by RC1 in Figure 3-127. Single transmit and receive

message buffers can be RC1-reconfigured when in the *HDis* or *HDisLck* state. Double transmit message buffers can be RC1-reconfigured if both the transmit side and the commit side are in the *HDis* state.

3.4.8.1.2 Buffer Type Not Changed (RC2)

A reconfiguration does not change the buffer type of the individual message buffer if the message buffer buffer type bit MBCCSRn.MBT is not changed. This type of reconfiguration is denoted by RC2 in Figure 3-127. It applies only to single transmit and receive message buffers. Single transmit and receive message buffers can be RC2-reconfigured when in the *HDis* or *HDisLck* state.

3.4.8.1.3 Buffer Type Changed (RC3)

A reconfiguration does change the buffer type of the individual message buffer if the message buffer type bit MBCCSRn.MBT is changed. This type of reconfiguration is denoted by RC3 in Figure 3-127. The RC3 reconfiguration splits one double buffer into two single buffers or combines two single buffer into one double buffer. In the later case, the two single message buffers must have consecutive message buffer numbers and the smaller one must be even. Message Buffers can be RC3 reconfigured if they are in the *HDis* state.

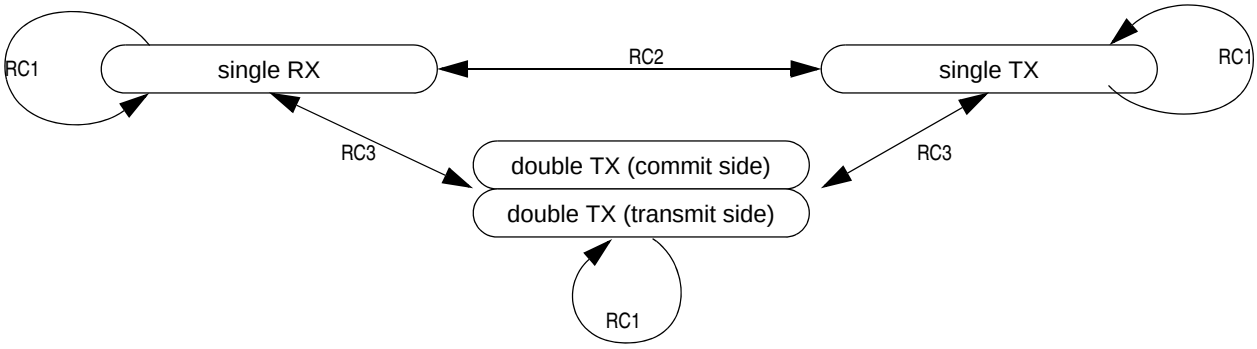


Figure 3-127. Message Buffer Reconfiguration Scheme

3.4.9 Receive FIFO

This section provides a detailed description of the two receive FIFOs.

3.4.9.1 Overview

The receive FIFOs implement the queued receive buffer defined by the *FlexRay Communications System Protocol Specification, Version 2.1 Rev A.* One receive FIFO is assigned to channel A, the other receive FIFO is assigned to channel B. Both FIFOs work completely independent from each other.

The message buffer structure of each FIFO is described in Section 3.4.3.3, “Receive FIFO”. The area in the FRM for each of the two receive FIFOs is characterized by:

- The index of the first FIFO entry given byReceive FIFO Start Index Register (RFSIR)
- The number of FIFO entries and the length of each FIFO entry as given by Receive FIFO Depth and Size Register (RFDSR)

3.4.9.2 Receive FIFO Configuration

The receive FIFO control and configuration data are given in Section 3.4.3.7, “Receive FIFO Control and Configuration Data”. The configuration of the receive FIFOs consists of two steps.

The first step is the allocation of the required amount of FRM for the FlexRay window. This includes the allocation of the message buffer header area and the allocation of the message buffer data fields. For more details see Section 3.4.4, “FlexRay Memory Layout”.

The second step is the programming of the configuration data register while the PE is in *POC:config.*

The following steps configure the layout of the FIFO.

- The number of the first message buffer header index that belongs to the FIFO is written into the Receive FIFO Start Index Register (RFSIR).
- The depth of the FIFO is written into the FIFO_DEPTH field in the Receive FIFO Depth and Size Register (RFDSR).
- The length of the message buffer data field for the FIFO is written into the ENTRY_SIZE field in the Receive FIFO Depth and Size Register (RFDSR).

NOTE

To ensure, that the read index RDIDX always points to a message buffer that contains valid data, the receive FIFO must have at least 2 entries.

The FIFO filters are configured through the fifo filter registers.

3.4.9.3 Receive FIFO Reception

The frame reception to the receive FIFO is enabled, if for a certain slots no message buffer is assigned or subscribed. In this case the FIFO filter path shown in Figure 3-128 is activated.

When the receive FIFO filter path indicates that the received frame must be appended to the FIFO, the FlexRay module writes the received frame header and slot status into the message buffer header field indicated by the internal FIFO header write index. The payload data are written in the message buffer data field. If the status of the received frame indicates a valid frame, the internal FIFO header write index is updated and the fifo not-empty interrupt flag FNEAIF/FNEBIF in the Global Interrupt Flag and Enable Register (GIFER) is set.

3.4.9.4 Receive FIFO Message Access

If the fifo not-empty interrupt flag FNEAIF/FNEBIF in the Global Interrupt Flag and Enable Register (GIFER) is set, the receive FIFO contains valid received messages, which can be accessed by the application.

The receive FIFO does not require locking to access the message buffers. To access the message the application first reads the receive FIFO read index RDIDX from the Receive FIFO A Read Index Register (RFARIR) or Receive FIFO B Read Index Register (RFBRIR), respectively. This index points to the message buffer header field of the next message buffer that contains valid data. The application can access the message data as described in Section 3.4.3.3, “Receive FIFO”. When the application has read all message buffer data and status information, it writes 1 to the fifo not-empty interrupt flags FNEAIF or

FNEBIF. This clears the interrupt flag and updates the RDIDX field in the Receive FIFO A Read Index Register (RFARIR) or Receive FIFO B Read Index Register (RFBRIR), respectively.When the RDIDX value has reached the last message buffer header field that belongs to the fifo, it wraps around to the index of the first message buffer header field that belongs to the fifo. This value is provided by the SIDX field in the Receive FIFO Start Index Register (RFSIR).

3.4.9.5 Receive FIFO filtering

The receive FIFO filtering is activated after all enabled individual receive message buffers have been searched without success for a message buffer to receive the current frame.

The FlexRay module provides three sets of FIFO filters. The FIFO filters are applied to valid non-null frames only. The FIFO does not receive invalid or null-frames. For each FIFO filter, the pass criteria is specified in the related section given below. Only frames that have passed all filters are appended to the FIFO. The FIFO filter path is depicted in Figure 3-128.

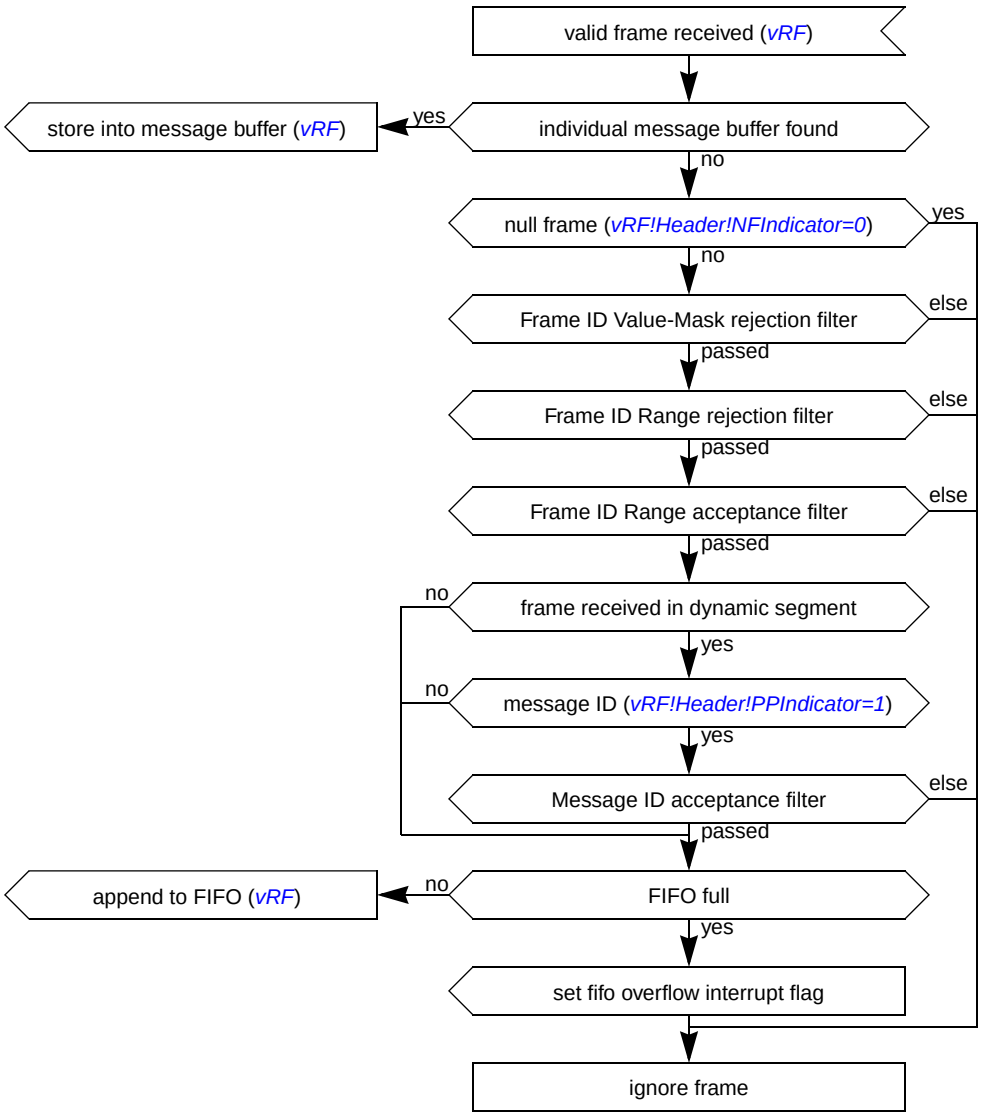


Figure 3-128. Received Frame FIFO Filter Path

A received frame passes the FIFO filtering if it has passed all three type of filter.

3.4.9.5.1 RX FIFO Frame ID Value-Mask Rejection Filter

The frame ID value-mask rejection filter is a value-mask filter and is defined by the fields in the Receive FIFO Frame ID Rejection Filter Value Register (RFFIDRFVR) and the Receive FIFO Frame ID Rejection Filter Mask Register (RFFIDRFMR). Each received frame with a frame ID FID that does not match the value-mask filter value passes the filter, i.e. is not rejected.

Consequently, a received valid frame with the frame ID FID passes the RX FIFO Frame ID Value-Mask Rejection Filter if Equation 3-10 is fulfilled.

$$'\mathrm{ID} \wedge \mathrm{RFFIDRFMR[FIDRFMSK]} \neq \mathrm{RFFIDRFVR[FIDRFVAL]} \wedge \mathrm{RFFIDRFMR[FIDRFMSK} \qquad \textit{Eqn. 3-10}$$

The RX FIFO Frame ID Value-Mask Rejection Filter can be configured to pass all frames by the following settings:

- RFFIDRFVR.FIDRFVAL:= 0x000
- RFFIDRFMR.FIDRFMSK:= 0x7FF

Using the settings above, only the frame with frame ID 0 is rejected, which is an invalid frame. All other frames pass.

The RX FIFO Frame ID Value-Mask Rejection Filter can be configured to reject all frames by the following settings.

- RFFIDRFMR.FIDRFMSK:= 0x000

Using the settings above, Equation 3-10 can never be fulfilled (0!= 0) and thus all frames are rejected; no frame pass. This is the reset value for the RX FIFO.

3.4.9.5.2 RX FIFO Frame ID Range Rejection Filter

Each of the four RX FIFO Frame ID Range filters can be configured as a rejection filter. The filters are configured by the Receive FIFO Range Filter Configuration Register (RFRFCFR) and controlled by the Receive FIFO Range Filter Control Register (RFRFCTR). The RX FIFO Frame ID range filters apply to all received valid frames. A received frame with the frame ID FID passes the RX FIFO Frame ID Range rejection filters if no rejection filter is enabled or, for all of the enabled RX FIFO Frame ID Range rejection filters, i.e. RFRFCTR.FiMD equals 1 and RFRFCTR.FiEN equals 1, Equation 3-11 is fulfilled.

$$\mathrm{FID} < \mathrm{RFRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=0}]) \text{ or } (\mathrm{RFRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=1}] < \mathrm{FID} \qquad \textit{Eqn. 3-11}$$

Consequently, all frames with a frame ID that fulfills Equation 3-12 for at least one of the enabled rejection filters is rejected and does not pass.

$$\wr\mathrm{FRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=0}] \le \mathrm{FID} \le \mathrm{RFRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=1} \qquad \textit{Eqn. 3-12}$$

3.4.9.5.3 RX FIFO Frame ID Range Acceptance filter

Each of the four RX FIFO Frame ID Range filters can be configured as an acceptance filter. The filters are configured by the Receive FIFO Range Filter Configuration Register (RFRFCFR) and controlled by the Receive FIFO Range Filter Control Register (RFRFCTR). The RX FIFO Frame ID range filters apply to all received valid frames. A received frame with the frame ID FID passes the RX FIFO Frame ID Range acceptance filters if no acceptance filter is enabled or, for at least one of the enabled RX FIFO Frame ID Range acceptance filters, i.e. RFRFCTR.FiMD equals 0 and RFRFCTR.FiEN equals 1, Equation 3-13 is fulfilled.

$$\wr\mathrm{FRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=0}] \le \mathrm{FID} \le \mathrm{RFRFCFR}_{\mathrm{SEL}}[\mathrm{SID}_{\mathrm{IBD}=1} \qquad \textit{Eqn. 3-13}$$

3.4.9.5.4 RX FIFO Message ID Acceptance Filter

The RX FIFO Message ID Acceptance Filter is a value-mask filter and is defined by the Receive FIFO Message ID Acceptance Filter Value Register (RFMIDAFVR) and the Receive FIFO Message ID Acceptance Filter Mask Register (RFMIAFMR). This filter applies only to valid frames received in the dynamic segment with the payload preamble indicator bit PPI set to 1. All other frames pass this filter.

A received valid frame in the dynamic segment with the payload preamble indicator bit PPI set to 1 and with the message ID MID (the first two bytes of the payload) passes the RX FIFO Message ID Acceptance Filter if Equation 3-14 is fulfilled.

$$MID \wedge RFMIDAFMR[MIDAFMSK] = RFMIDAFMR[MIDAFVAL] \wedge RFMIDAFMR[MIDAFMSK] \quad \textit{Eqn. 3-14}$$

The RX FIFO Message ID Acceptance Filter can be configured to accept all frames by setting

- RFMIDAFMR.MIDAFMSK:= 0x000

Using the settings above, Equation 3-14 is always fulfilled and all frames pass.

3.4.10 Channel Device Modes

This section describes the two FlexRay channel device modes that are supported by the FlexRay module.

3.4.10.1 Dual Channel Device Mode

In the dual channel device mode, both FlexRay ports are connected to physical FlexRay bus lines. The FlexRay port consisting of RXD_BG1, TXD_BG1, and TXEN1# is connected to the physical bus channel A and the FlexRay port consisting of RXD_BG2, TXD_BG2, and TXEN1# is connected to the physical bus channel B. The dual channel system is shown in Figure 3-129.

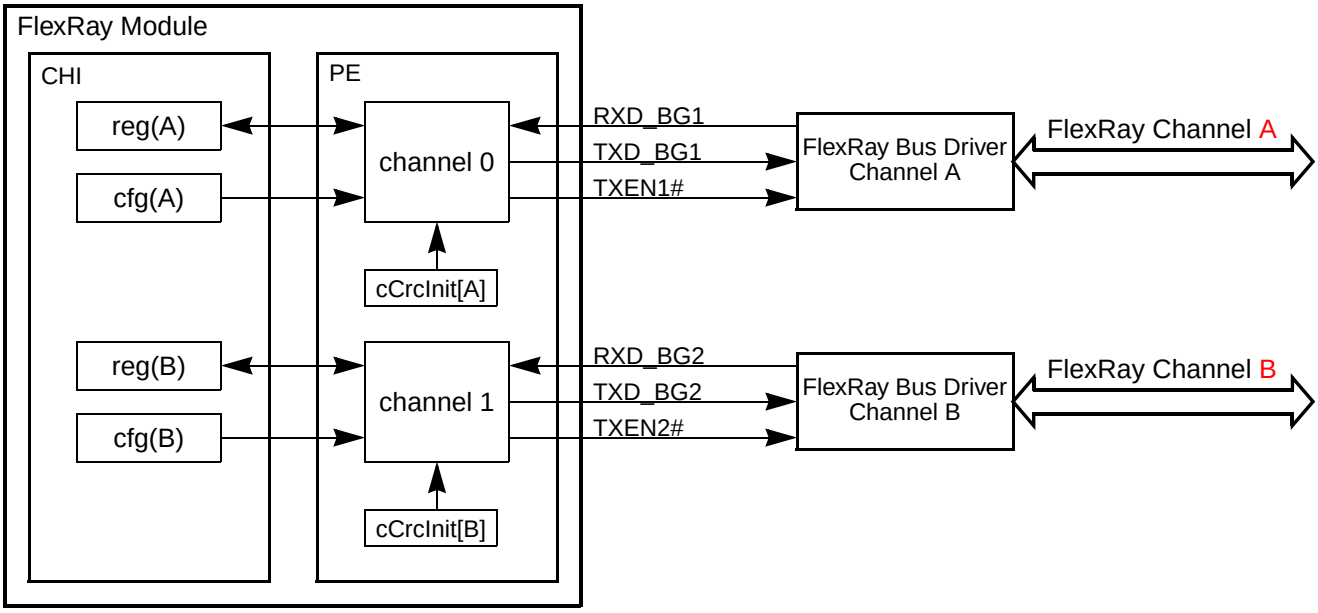


Figure 3-129. Dual Channel Device Mode

3.4.10.2 Single Channel Device Mode

The single channel device mode supports devices that have only one FlexRay port available. This FlexRay port consists of the signals RXD_BG1, TXD_BG1, and TXEN1# and can be connected to the physical bus channel A (shown in Figure 3-130) or the physical bus channel B (shown in Figure 3-131).

If the device is configured as a single channel device by setting MCR.SCD to 1, only the internal channel A and the FlexRay Port A is used. Depending on the setting of MCR.CHA and MCR.CHB, the internal channel A behaves as a FlexRay Channel A or FlexRay Channel B. The bit MCR.CHA must be set, if the FlexRay Port A is connected to a FlexRay Channel A. The bit MCR.CHB must be set if the FlexRay Port A is connected to a FlexRay Channel B. The two FlexRay channels differ only in the initial value for the frame CRC *cCrcInit*. For a single channel device, the application can access and configure only the registers related to internal channel A.

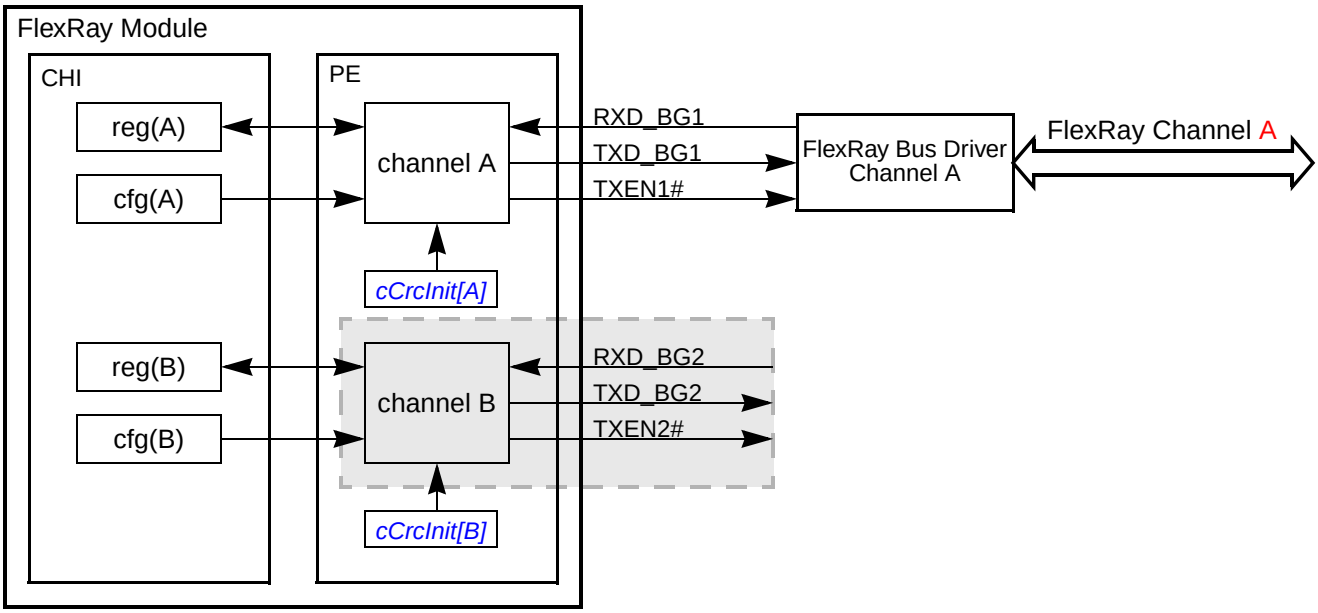


Figure 3-130. Single Channel Device Mode (Channel A)

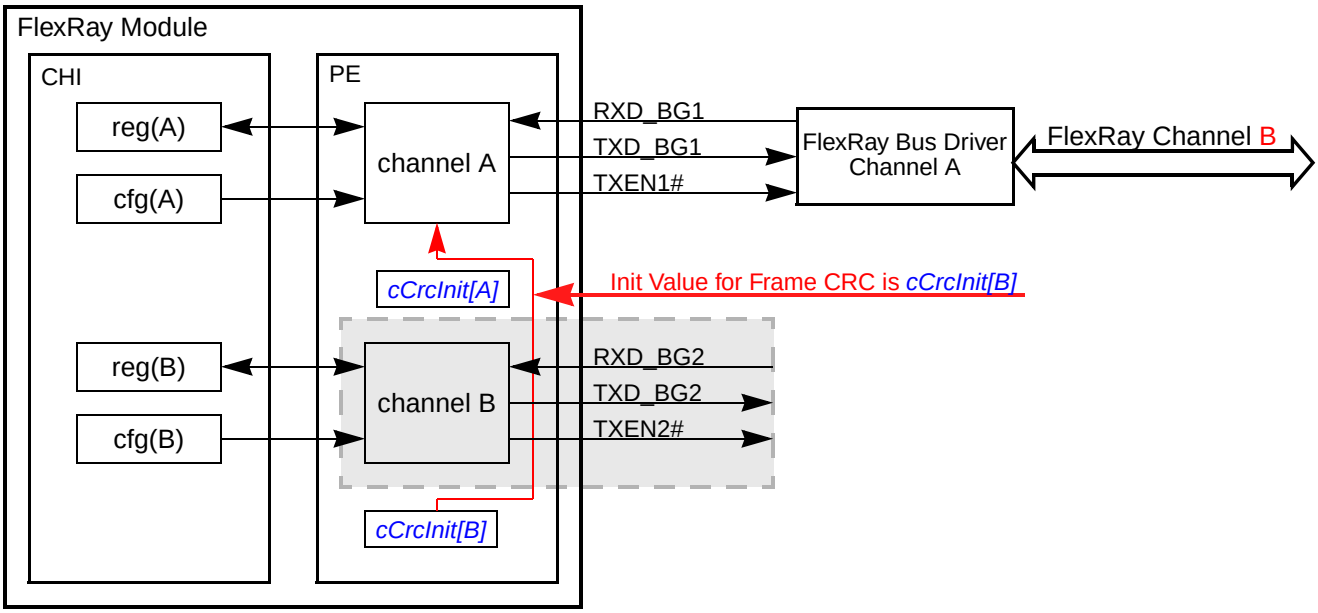


Figure 3-131. Single Channel Device Mode (Channel B)

3.4.11 External Clock Synchronization

The application of the external rate and offset correction is triggered when the application writes to the EOC_AP and ERC_AP fields in the Protocol Operation Control Register (POCR). The PE applies the external correction values in the next even-odd cycle pair as shown in Figure 3-132 and Figure 3-133.

NOTE

The values provided in the EOC_AP and ERC_AP fields are the values that were written from the application most recently. If these value were already applied, they are not applied in the current cycle pair again.

If the offset correction applied in the NIT of cycle 2n+1 shall be affect by the external offset correction, the EOC_AP field must be written to after the start of cycle 2n and before the end of the static segment of cycle 2n+1. If this field is written to after the end of the static segment of cycle 2n+1, it is not guaranteed that the external correction value is applied in cycle 2n+1. If the value is not applied in cycle 2n+1, the value is applied in the cycle 2n+3. Refer to Figure 3-132 for timing details.

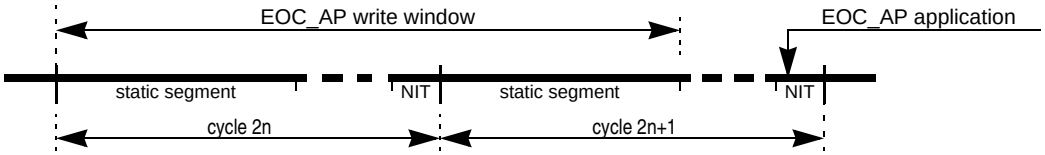


Figure 3-132. External Offset Correction Write and Application Timing

If the rate correction for the cycle pair [2n+2, 2n+3] shall be affect by the external offset correction, the ERC_AP field must be written to after the start of cycle 2n and before the end of the static segment start of cycle 2n+1. If this field is written to after the end of the static segment of cycle 2n+1, it is not guaranteed that the external correction value is applied in cycle pair [2n+2, 2n+3]. If the value is not applied for cycle pair [2n+2, 2n+3], the value is be applied for cycle pair [2n+4, 2n+5]. Refer to Figure 3-133 for details.

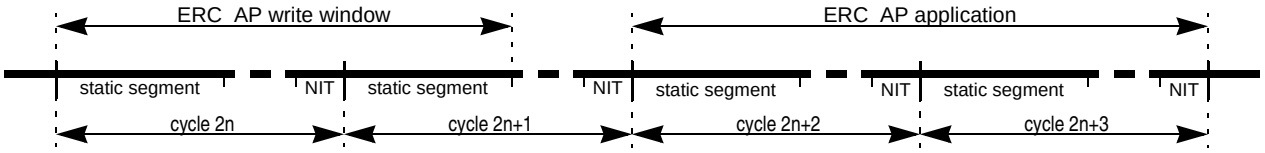


Figure 3-133. External Rate Correction Write and Application Timing

3.4.12 Sync Frame ID and Sync Frame Deviation Tables

The FlexRay protocol requires the provision of a snapshot of the Synchronization Frame ID tables for the even and odd communication cycle for both channels. The FlexRay module provides the means to write a copy of these internal tables into the FRM and ensures application access to consistent tables by means of table locking. After the application has locked the table successfully, the FlexRay module does not overwrite these tables and the application can read a consistent snapshot.

NOTE

Only synchronization frames that have passed the synchronization frame filters are considered for clock synchronization and appear in the sync frame tables.

3.4.12.1 Sync Frame ID Table Content

The Sync Frame ID Table is a snapshot of the protocol related variables *vsSyncIdListA* and *vsSyncIdListB* for each even and odd communication cycle. This table provides a list of the frame IDs of the synchronization frames received on the corresponding channel and cycle that are used for the clock synchronization.

3.4.12.2 Sync Frame Deviation Table Content

The Sync Frame Deviation Table is a snapshot of the protocol related variable zsDev(id)(oe)(ch)!Value. Each Sync Frame Deviation Table entry provides the deviation value for the sync frame, with the frame ID presented in the corresponding entry in the Sync Frame ID Table.

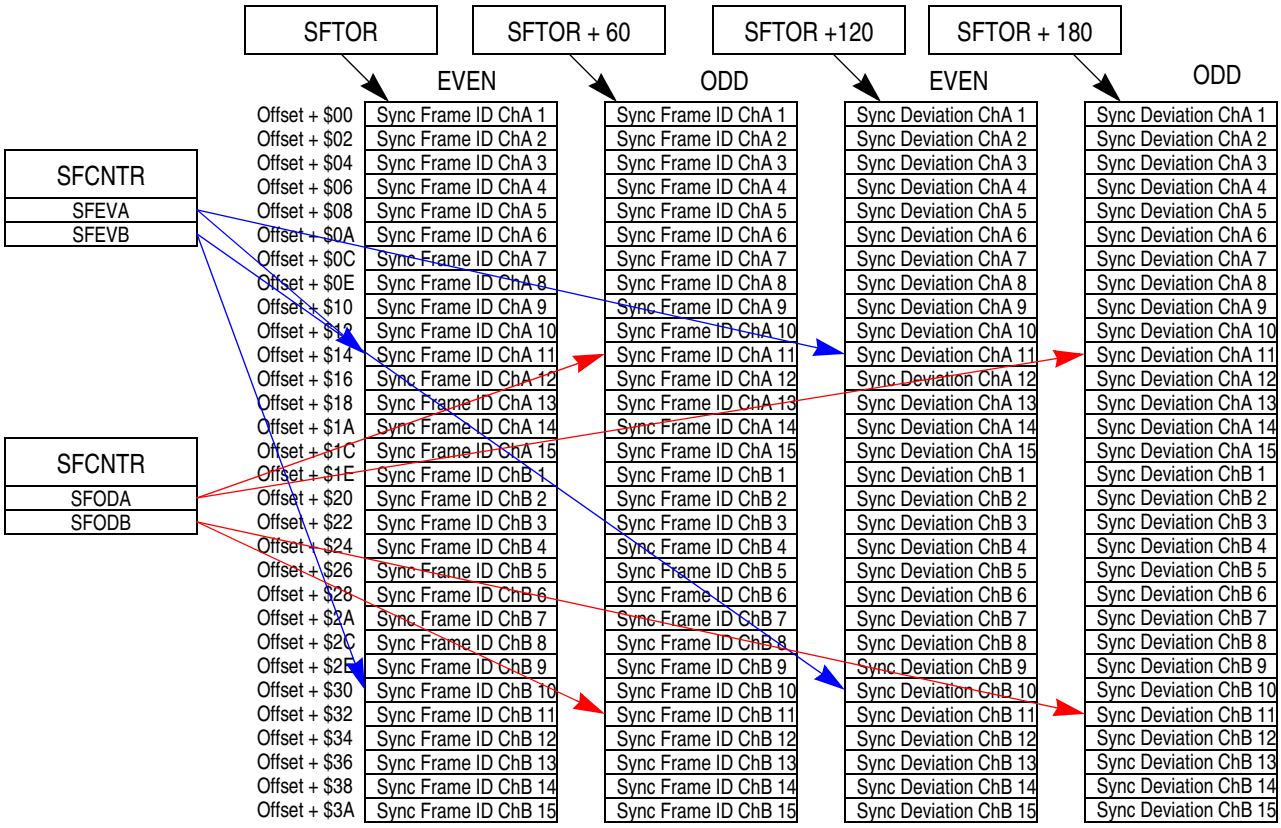


Figure 3-134. Sync Table Memory Layout

3.4.12.3 Sync Frame ID and Sync Frame Deviation Table Setup

The FlexRay module writes a copy of the internal synchronization frame ID and deviation tables into the FRM if requested by the application. The application must provide the appropriate amount of FRM for the tables. The memory layout of the tables is given in Figure 3-134. Each table occupies 120 16-bit entries.

While the protocol is in *POC:config* state, the application must program the offsets for the tables into the Sync Frame Table Offset Register (SFTOR).

3.4.12.4 Sync Frame ID and Sync Frame Deviation Table Generation

The application controls the generation process of the Sync Frame ID and Sync Frame Deviation Tables into the FRM using the Sync Frame Table Configuration, Control, Status Register (SFTCCSR). A summary of the copy modes is given in Table 3-108.

Table 3-108. Sync Frame Table Generation Modes

SFTCCSR			Description
OPT	SDVEN	SIDEN	
0	0	0	No Sync Frame Table copy
0	0	1	Sync Frame ID Tables are copied continuously
0	1	0	Reserved
0	1	1	Sync Frame ID Tables and Sync Frame Deviation Tables are copied continuously
1	0	0	No Sync Frame Table copy
1	0	1	Sync Frame ID Tables for next even-odd-cycle pair are copied
0	1	0	Reserved
1	1	1	Sync Frame ID Tables and Sync Frame Deviation Tables for next even-odd-cycle pair are copied

The Sync Frame Table generation process is described in the following for the even cycle. The same sequence applies to the odd cycle.

If the application has enabled the sync frame table generation by setting SFTCCSR.SIDEN to 1, the FlexRay module starts the update of the even cycle related tables after the start of the NIT of the next even cycle. The FlexRay module checks if the application has locked the tables by reading the SFTCCSR.ELKS lock status bit. If this bit is set, the FlexRay module does not update the table in this cycle. If this bit is cleared, the FlexRay module locks this table and starts the table update. To indicate that these tables are currently updated and may contain inconsistent data, the FlexRay module clears the even table valid status bit SFTCCSR.EVAL. After all table entries related to the even cycle have been transferred into the FRM, the FlexRay module sets the even table valid bit SFTCCSR.EVAL and the Even Cycle Table Written Interrupt Flag EVT_IF in the Protocol Interrupt Flag Register 1 (PIFR1). If the interrupt enable flag EVT_IE is set, an interrupt request is generated.

To read the generated tables, the application must lock the tables to prevent the FlexRay module from updating these tables. The locking is initiated by writing a 1 to the even table lock trigger SFTCCSR.ELKT. When the even table is not currently updated by the FlexRay module, the lock is granted and the even table lock status bit SFTCCSR.ELKS is set. This indicates that the application has successfully locked the even sync tables and the corresponding status information fields SFRA, SFRB in

the Sync Frame Counter Register (SFCNTR). The value in the SFTCCSR.CYCNUM field provides the number of the cycle that this table is related to.

The number of available table entries per channel is provided in the SFCNTR.SFEVA and SFCNTR.SFEVB fields. The application can now start to read the sync table data from the locations given in Figure 3-134.

After reading all the data from the locked tables, the application must unlock the table by writing to the even table lock trigger SFTCCSR.ELKT again. The even table lock status bit SFTCCSR.ELKS is reset immediately.

If the sync frame table generation is disabled, the table valid bits SFTCCSR.EVAL and SFTCCSR.EVAL are reset when the counter values in the Sync Frame Counter Register (SFCNTR) are updated. This is done because the tables stored in the FRM are no longer related to the values in the Sync Frame Counter Register (SFCNTR).

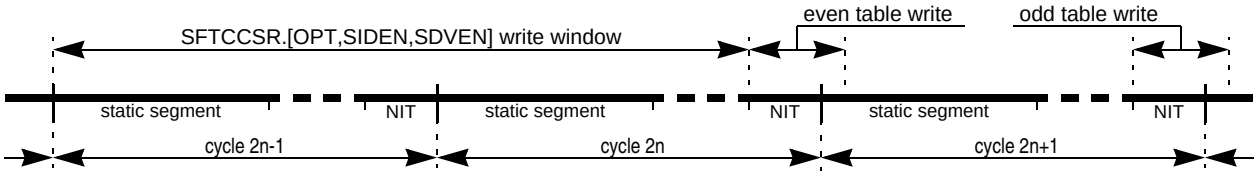


Figure 3-135. Sync Frame Table Trigger and Generation Timing

3.4.12.5 Sync Frame Table Access

The sync frame tables is transferred into the FRM during the table write windows shown in Figure 3-135. During the table write, the application can not lock the table that is currently written. If the application locks the table outside of the table write window, the lock is granted immediately.

3.4.12.5.1 Sync Frame Table Locking and Unlocking

The application locks the even/odd sync frame table by writing 1 to the lock trigger bit ELKT/OLKT in the Sync Frame Table Configuration, Control, Status Register (SFTCCSR). If the affected table is not currently written to the FRM, the lock is granted immediately, and the lock status bit ELKS/OLKS is set. If the affected table is currently written to the FRM, the lock is not granted. In this case, the application must issue the lock request again until the lock is granted.

The application unlocks the even/odd sync frame table by writing 1 to the lock trigger bit ELKT/OLKT. The lock status bit ELKS/OLKS is cleared immediately.

3.4.13 MTS Generation

The FlexRay module provides a flexible means to request the transmission of the Media Access Test Symbol MTS in the symbol window on channel A or channel B.

The application can configure the set of communication cycles in which the MTS is transmitted over the FlexRay bus by programming the CYCCNTMSK and CYCCNTVAL fields in the MTS A Configuration Register (MTSACFR) and MTS B Configuration Register (MTSBCFR).

The application enables or disables the generation of the MTS on either channel by setting or clearing the MTE control bit in the MTS A Configuration Register (MTSACFR) or MTS B Configuration Register (MTSBCFR). If an MTS is to be transmitted in a certain communication cycle, the application must set the MTE control bit during the static segment of the preceding communication cycle.

The MTS is transmitted over channel A in the communication cycle with number CYCCNT, if Equation 3-16, Equation 3-17, and Equation 3-17 are fulfilled.

$$PSR0[PROTSTATE] = \textit{POC:normal active} \qquad \textit{Eqn. 3-15}$$

$$MTSACRF[MTE] = 1 \qquad \textit{Eqn. 3-16}$$

$$CYCCNT \wedge MTSACFR[CYCCNTMSK] = MTSACFR[CYCCNTVAL] \wedge MTSACFR[CYCCNTMSK] \qquad \textit{Eqn. 3-17}$$

The MTS is transmitted over channel B in the communication cycle with number CYCCNT, if Equation 3-15, Equation 3-18, and Equation 3-19 are fulfilled.

$$MTSBCRF[MTE] = 1 \qquad \textit{Eqn. 3-18}$$

$$CYCCNT \wedge MTSBCFR[CYCCNTMSK] = MTSBCFR[CYCCNTVAL] \wedge MTSBCFR[CYCCNTMSK] \qquad \textit{Eqn. 3-19}$$

3.4.14 Sync Frame and Startup Frame Transmission

The transmission of sync frames and startup frames is controlled by the following register fields:

- PCR18.key_slot_id: provides the number of the slot for sync or startup frame transmission
- PCR11.key_slot_used_for_sync: indicates sync frame transmission
- PCR11.key_slot_used_for_startup: indicates startup frame transmission
- PCR12.key_slot_header_crc: provides header crc for sync frame or startup frame
- Message buffer with message buffer number n=PCR18.key_slot_id

The generation of the sync or startup frames depends on the current protocol state. In the *POC:startup* state, the generation is independent of the message buffer setup; in the *POC:normal active* state, the generation is affected by the current message buffer setup.

3.4.14.1 Sync Frame and Startup Frame Transmission in *POC:startup*

In the *POC:startup* state, the sync and startup frame transmission is independent of the message buffer setup. If at least one of the indication bits PCR11.key_slot_used_for_sync or PCR11.key_slot_used_for_startup is set, a Null Frame is transmitted in the slot with slot number PCR18.key_slot_id. The header CRC for this Null Frame is taken from PCR12.key_slot_header_crc. The settings of the sync and startup frame indicators are taken from PCR11.key_slot_used_for_sync and PCR11.key_slot_used_for_startup.

3.4.14.2 Sync Frame and Startup Frame Transmission in *POC:normal active*

In the *POC:normal active* state, the sync and startup frame transmission depends on the message buffer setup. If at least one of the indication bits PCR11.key_slot_used_for_sync or

PCR11.key_slot_used_for_startup is set, or if a transmit message buffer with MBFIDRn.FID == PCR18.key_slot_id is configured and enabled, a Null Frame or Data Frame is transmitted in the slot with slot number PCR18.key_slot_id. The header CRC for this frame is taken from PCR12.key_slot_header_crc, the settings of the sync and startup frame indicators are taken from PCR11.key_slot_used_for_sync and PCR11.key_slot_used_for_startup. A data frame is transmitted if the message buffer is unlocked and committed and the cycle counter filter matches the current cycle.

3.4.15 Sync Frame Filtering

Each received synchronization frame must pass the Sync Frame Acceptance Filter and the Sync Frame Rejection Filter before it is considered for clock synchronization. If the synchronization frame filtering is globally disabled, i.e. the SFFE control bit in the Module Configuration Register (MCR) is cleared, all received synchronization frames are considered for clock synchronization. If a received synchronization frame did not pass at least one of the two filters, this frame is processed as a normal frame and is not considered for clock synchronization.

3.4.15.1 Sync Frame Acceptance Filtering

The synchronization frame acceptance filter is implemented as a value-mask filter. The value is configured in the Sync Frame ID Acceptance Filter Value Register (SFIDAFVR) and the mask is configured in the Sync Frame ID Acceptance Filter Mask Register (SFIDAFMR). A received synchronization frame with the frame ID FID passes the sync frame acceptance filter, if Equation 3-20 or Equation 3-21evaluates to true.

$$MCR[SFFE] = 0 \qquad \textit{Eqn. 3-20}$$

$$FID[9:0] \wedge SFIDAFMR[FMSK[9:0]] = SFIDAFVR[FVAL[9:0]] \wedge SFIDAFMR[FMSK[9:0] \qquad \textit{Eqn. 3-21}$$

NOTE

Sync frames are transmitted in the static segment only. Thus FID is less than or equal to 1023.

3.4.15.2 Sync Frame Rejection Filtering

The synchronization frame rejection filter is a comparator. The compare value is defined by the Sync Frame ID Rejection Filter Register (SFIDRFR). A received synchronization frame with the frame ID FID passes the sync frame rejection filter if Equation 3-22 or Equation 3-23 evaluates to true.

$$MCR[SFFE] = 0 \qquad \textit{Eqn. 3-22}$$

$$FID[9:0] \neq SFIDRFR[SYNFRID[9:0]] \qquad \textit{Eqn. 3-23}$$

NOTE

Sync frames are transmitted in the static segment only. Thus FID is less than or equal to 1023.

3.4.16 Strobe Signal Support

The FlexRay module provides a number of strobe signals for observing internal protocol timing related signals in the protocol engine. The signals are listed and described in Table 3-12.

3.4.16.1 Strobe Signal Assignment

Each of the strobe signals listed in Table 3-12 can be assigned to one of the four strobe ports using the Strobe Signal Control Register (STBSCR). To assign multiple strobe signals, the application must write multiple times to the Strobe Signal Control Register (STBSCR) with appropriate settings.

To read out the current settings for a strobe signal with number N, the application must execute the following sequence:

1. Write to STBSCR with WMD equaling 1 and SEL equaling N. (updates SEL field only)
2. Read STBCSR.

   The SEL field provides N and the ENB and STBPSEL fields provides the settings for signal N.

3.4.16.2 Strobe Signal Timing

This section provides detailed timing information of the strobe signals with respect to the protocol engine clock.

The strobe signals display internal PE signals. Due to the internal architecture of the PE, some signals are generated several PE clock cycles before the actual action is performed on the FlexRay Bus. These signals are listed in Table 3-12 with a negative clock offset. An example waveform is given in Figure 3-136.

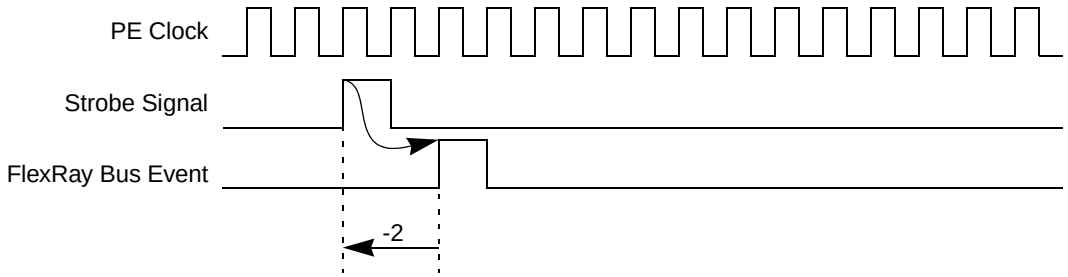


Figure 3-136. Strobe Signal Timing (type = pulse, clk_offset = -2)

Other signals refer to events that occurred on the FlexRay Bus some cycles before the strobe signal is changed. These signals are listed in Table 3-12 with a positive clock offset. An example waveform is given in Figure 3-137.

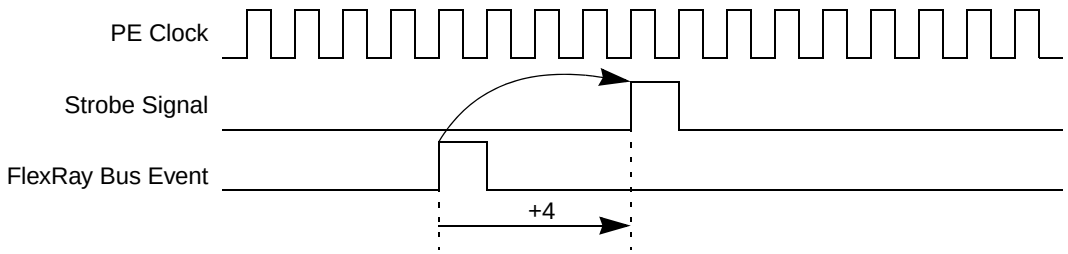


Figure 3-137. Strobe Signal Timing (type = pulse, clk_offset = +4)

3.4.17 Timer Support

The FlexRay module provides two timers, which run on the FlexRay time base. Each timer generates a maskable interrupt when it reaches a configured point in time. Timer T1 is an absolute timer. Timer T2 can be configured to be an absolute or a relative timer. Both timers can be configured to be repetitive. In the non-repetitive mode, timer stops if it expires. In repetitive mode, timer is restarted when it expires.

Both timers are active only when the protocol is in *POC:normal active* or *POC:normal passive* state. If the protocol is not in one of these modes, the timers are stopped. The application must restart the timers when the protocol has reached the *POC:normal active* or *POC:normal passive* state.

3.4.17.1 Absolute Timer T1

The absolute timer T1 has the protocol cycle count and the macrotick count as the time base. The timer 1 interrupt flag TI1_IF in the Protocol Interrupt Flag Register 0 (PIFR0) is set at the macrotick start event, if Equation 3-24 and Equation 3-25 are fulfilled

CYCCTR.CYCCNT & T1CYSR.T1_CYC_MSK == T1CYSR.T1_CYC_VAL & T1CYSR.T1_CYC_MSK *Eqn. 3-24*

MTCTR.MTCT == TI1MTOR.T1_MTOFFSET *Eqn. 3-25*

If the timer 1 interrupt enable bit TI1_IE in the Protocol Interrupt Enable Register 0 (PIER0) is asserted, an interrupt request is generated.

The status bit T1ST is set when the timer is triggered, and is cleared when the timer expires and is non-repetitive. If the timer expires but is repetitive, the T1ST bit is not cleared and the timer is restarted immediately. The T1ST is cleared when the timer is stopped.

3.4.17.2 Absolute / Relative Timer T2

The timer T2 can be configured to be an absolute or relative timer by setting the T2_CFG control bit in the Timer Configuration and Control Register (TICCR). The status bit T2ST is set when the timer is triggered, and is cleared when the timer expires and is non-repetitive. If the timer expires but is repetitive, the T2ST bit is not cleared and the timer is restarted immediately. The T2ST is cleared when the timer is stopped.

3.4.17.2.1 Absolute Timer T2

If timer T2 is configured as an absolute timer, it has the same functionality timer T1 but the configuration from Timer 2 Configuration Register 0 (TI2CR0) and Timer 2 Configuration Register 1 (TI2CR1) is used. On expiration of timer T2, the interrupt flag TI2_IF in the Protocol Interrupt Flag Register 0 (PIFR0) is set. If the timer 1 interrupt enable bit TI1_IE in the Protocol Interrupt Enable Register 0 (PIER0) is asserted, an interrupt request is generated.

3.4.17.2.2 Relative Timer T2

If the timer T2 is configured as a relative timer, the interrupt flag TI2_IF in the Protocol Interrupt Flag Register 0 (PIFR0) is set, when the programmed amount of macroticks MT[31:0], defined by Timer 2 Configuration Register 0 (TI2CR0) and Timer 2 Configuration Register 1 (TI2CR1), has expired since the trigger or restart of timer 2. The relative timer is implemented as a down counter and expires when it has

reached 0. At the macrotick start event, the value of MT[31:0] is checked and then decremented. Thus, if the timer is started with MT[31:0] == 0, it expires at the next macrotick start.

3.4.18 Slot Status Monitoring

The FlexRay module provides several means for slot status monitoring. All slot status monitors use the same slot status vector provided by the PE. The PE provides a slot status vector for each static slot, for each dynamic slot, for the symbol window, and for the NIT, on a per channel base. The content of the slot status vector is described in Table 3-109. The PE provides the slot status vector within the first macrotick after the end of the related slot/window/NIT, as shown in Figure 3-138.

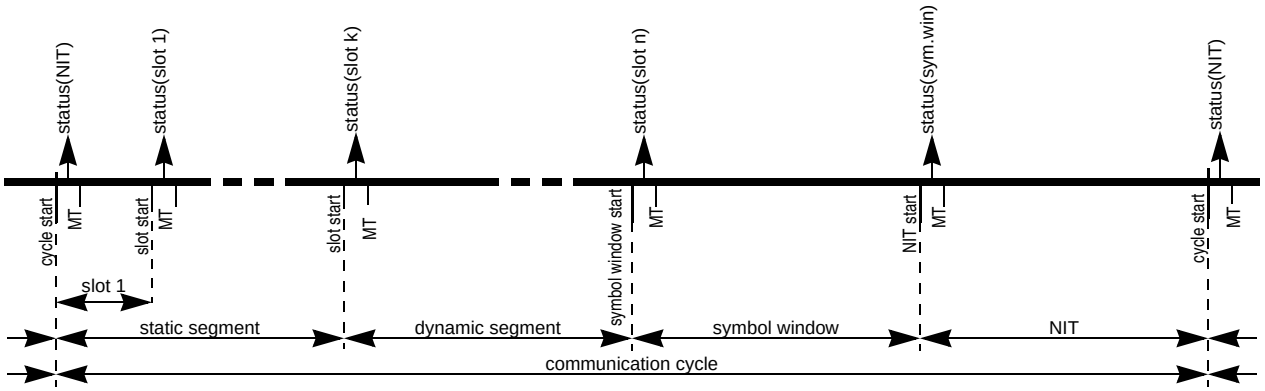


Figure 3-138. Slot Status Vector Update

NOTE

The slot status for the NIT of cycle n is provided after the start of cycle n+1.

Table 3-109. Slot Status Content

	Status Content
static / dynamic Slot	**slot related status** *vSS!ValidFrame* - valid frame received *vSS!SyntaxError* - syntax error occurred while receiving *vSS!ContentError* - content error occurred while receiving *vSS!BViolation* - boundary violation while receiving *for slots in which the module transmits:* *vSS!TxConflict* - reception ongoing while transmission starts *for slots in which the module does not transmit:* *vSS!TxConflict* - reception ongoing while transmission starts first valid - channel that has received the first valid frame **received frame related status** *extracted from* *a) header of valid frame, if vSS!ValidFrame = 1* *b) last received header, if vSS!ValidFrame = 0* *c) set to 0, if nothing was received* *vRF!Header!NFIndicator* - Null Frame Indicator (0 for null frame) *vRF!Header!SuFIndicator* - Startup Frame Indicator *vRF!Header!SyFIndicator* - Sync Frame Indicator
Symbol Window	**window related status** *vSS!ValidFrame* - always 0 *vSS!ContentError* - content error occurred while receiving *vSS!SyntaxError* - syntax error occurred while receiving *vSS!BViolation* - boundary violation while receiving *vSS!TxConflict* - reception ongoing while transmission starts **received symbol related status** *vSS!ValidMTS* - valid Media Test Access Symbol received **received frame related status** *see static/dynamic slot*
NIT	**NIT related status** *vSS!ValidFrame* - always 0 *vSS!ContentError* - content error occurred while receiving *vSS!SyntaxError* - syntax error occurred while receiving *vSS!BViolation* - boundary violation while receiving *vSS!TxConflict* - always 0 **received frame related status** *see static/dynamic slot*

3.4.18.1 Channel Status Error Counter Registers

The two channel status error counter registers, Channel A Status Error Counter Register (CASERCR) and Channel B Status Error Counter Register (CBSERCR), incremented by one, if at least one of four slot status error bits, *vSS!SyntaxError*, *vSS!ContentError*, *vSS!BViolation*, or *vSS!TxConflict* is set to 1. The status vectors for all slots in the static and dynamic segment, in the symbol window, and in the NIT are taken into account. The counters wrap round after they have reached the maximum value.

3.4.18.2 Protocol Status Registers

The Protocol Status Register 2 (PSR2) provides slot status information about the Network Idle Time NIT and the Symbol Window. The Protocol Status Register 3 (PSR3) provides aggregated slot status information.

3.4.18.3 Slot Status Registers

The eight slot status registers, Slot Status Registers (SSR0–SSR7), can be used to observe the status of static slots, dynamic slots, the symbol window, or the NIT without individual message buffers. These registers provide all slot status related and received frame / symbol related status information, as given in Table 3-109, except of the *first valid* indicator for non-transmission slots.

3.4.18.4 Slot Status Counter Registers

The FlexRay module provides four slot status error counter registers, Slot Status Counter Registers (SSCR0–SSCR3). Each of these slot status counter registers is updated with the value of an internal slot status counter at the start of a communication cycle. The internal slot status counter is incremented if its increment condition, defined by the Slot Status Counter Condition Register (SSCCR), matches the status vector provided by the PE. All static slots, the symbol window, and the NIT status are taken into account. *Dynamic* slots are *excluded*. The internal slot status counting and update timing is shown in Figure 3-139.

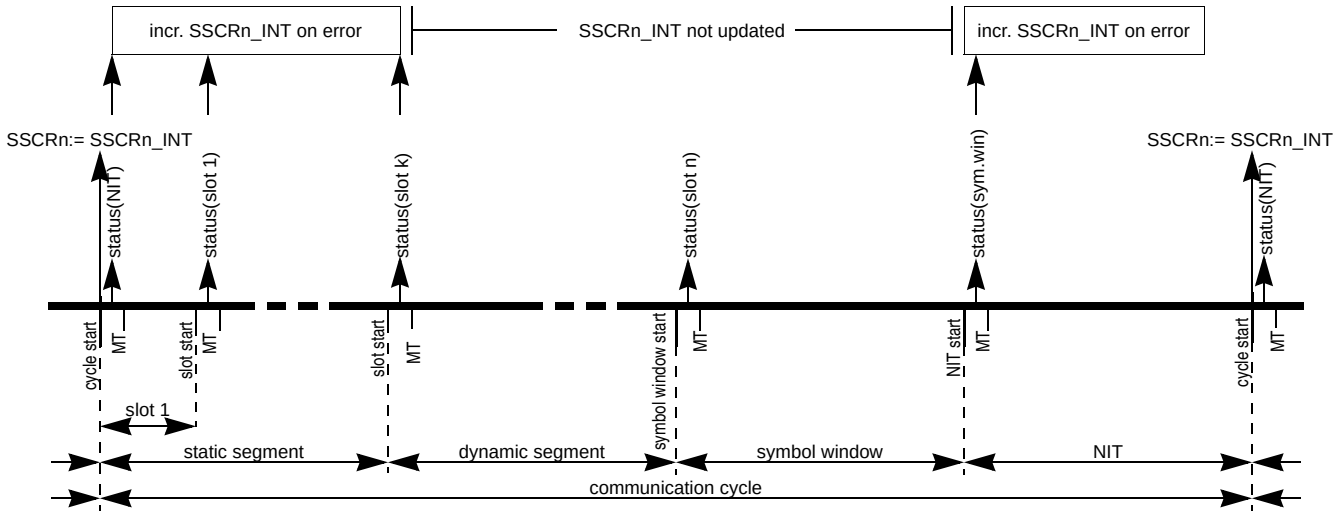


Figure 3-139. Slot Status Counting and SSCRn Update

The PE provides the status of the NIT in the first slot of the next cycle. Due to these facts, the SSCRn register reflects, in cycle n, the status of the NIT of cycle n-2, and the status of all static slots and the symbol window of cycle n-1.

The increment condition for each slot status counter consists of two parts, the frame related condition part and the slot related condition part. The internal slot status counter SSCRn_INT is incremented if at least one of the conditions is fulfilled:

1. frame related condition:

- (SSCCRn.VFR | SSCCRn.SYF | SSCCRn.NUF| SSCCRn.SUF) // count on frame condition = 1;

and

- ((~SSCCRn.VFR |*vSS!ValidFrame*) & // valid frame restriction
  (~SSCCRn.SYF | *vRF!Header!SyFIndicator*) & // sync frame indicator restriction
  (~SSCCRn.NUF | ~*vRF!Header!NFIndicator*) & // null frame indicator restriction
  (~SSCCRn.SUF | *vRF!Header!SuFIndicator*)) // startup frame indicator restriction
  = 1;

NOTE

The indicator bits SYF, NUF, and SUF are valid only when a valid frame was received. Thus it is required to set the VFR always, when count on frame condition is used.

2. slot related condition:

- ((SSCCRn.STATUSMASK[3] &*vSS!ContentError*) | // increment on content error
  (SSCCRn.STATUSMASK[2] & *vSS!SyntaxError*) | // increment on syntax error
  (SSCCRn.STATUSMASK[1] & *vSS!BViolation*) | // increment on boundary violation
  (SSCCRn.STATUSMASK[0] & *vSS!TxConflict*)) // increment on transmission conflict
  = 1;

If the slot status counter is in single cycle mode, i.e. SSCCRn.MCY equals 0, the internal slot status counter SSCRn_INT is reset at each cycle start. If the slot status counter is in the multicycle mode, i.e. SSCCRn.MCY equals 1, the counter is not reset and incremented, until the maximum value is reached.

3.4.18.5 Message Buffer Slot Status Field

Each individual message buffer and each FIFO message buffer provides a slot status field, which provides the information shown in Table 3-109 for the static/dynamic slot. The update conditions for the slot status field depend on the message buffer type. Refer to the Message Buffer Update Sections in Section 3.4.6, “Individual Message Buffer Functional Description”.

3.4.19 Interrupt Support

The FlexRay module provides 172 individual interrupt sources and five combined interrupt sources.

3.4.19.1 Individual Interrupt Sources

3.4.19.1.1 Message Buffer Interrupts

The FlexRay module provides 128 message buffer interrupt sources.

Each individual message buffer provides an interrupt flag MBCCSn.MBIF and an interrupt enable bit MBCCSn.MBIE. The FlexRay module sets the interrupt flag when the slot status of the message buffer was updated. If the interrupt enable bit is asserted, an interrupt request is generated.

3.4.19.1.2 Receive FIFO Interrupts

The FlexRay module provides 2 Receive FIFO interrupt sources.

Each of the 2 Receive FIFO provides a Receive FIFO Not Empty Interrupt Flag. The FlexRay module sets the Receive FIFO Not Empty Interrupt Flags (GIFER.FNEBIF, GIFER.FNEAIF) in the Global Interrupt Flag and Enable Register (GIFER) if the corresponding Receive FIFO is not empty.

3.4.19.1.3 Wakeup Interrupt

The FlexRay module provides one interrupt source related to the wakeup.

The FlexRay module sets the Wakeup Interrupt Flag GIFER.WUPIF when it has received a wakeup symbol on the FlexRay bus. The FlexRay module generates an interrupt request if the interrupt enable bit GIFER.WUPIE is asserted.

3.4.19.1.4 Protocol Interrupts

The FlexRay module provides 25 interrupt sources for protocol related events. For details, see Protocol Interrupt Flag Register 0 (PIFR0) and Protocol Interrupt Flag Register 1 (PIFR1). Each interrupt source has its own interrupt enable bit.

3.4.19.1.5 CHI Error Interrupts

The FlexRay module provides 16 interrupt sources for CHI related error events. For details, see CHI Error Flag Register (CHIERFR). There is one common interrupt enable bit GIFER.CHIIE for all CHI error interrupt sources.

3.4.19.2 Combined Interrupt Sources

Each combined interrupt source generates an interrupt request only when at least one of the interrupt sources that is combined generates an interrupt request.

3.4.19.2.1 Receive Message Buffer Interrupt

The combined receive message buffer interrupt request RBIRQ is generated when at least one of the individual receive message buffers generates an interrupt request MBXIRQ[n] and the interrupt enable bit GIFER.RBIE is set.

3.4.19.2.2 Transmit Message Buffer Interrupt

The combined transmit message buffer interrupt request TBIRQ is generated when at least one of the individual transmit message buffers generates an interrupt request MBXIRQ[n] and the interrupt enable bit GIFER.TBIE is asserted.

3.4.19.2.3 Protocol Interrupt

The combined protocol interrupt request PRTIRQ is generated when at least one of the individual protocol interrupt sources generates an interrupt request and the interrupt enable bit GIFER.PRIE is set.

3.4.19.2.4 CHI Error Interrupt

The combined CHI error interrupt request CHIIRQ is generated when at least one of the individual chi error interrupt sources generates an interrupt request and the interrupt enable bit GIFER.CHIE is set.

3.4.19.2.5 Module Interrupt

The combined module interrupt request MIRQ is generated if at least one of the combined interrupt sources generates an interrupt request and the interrupt enable bit GIFER.MIE is set.

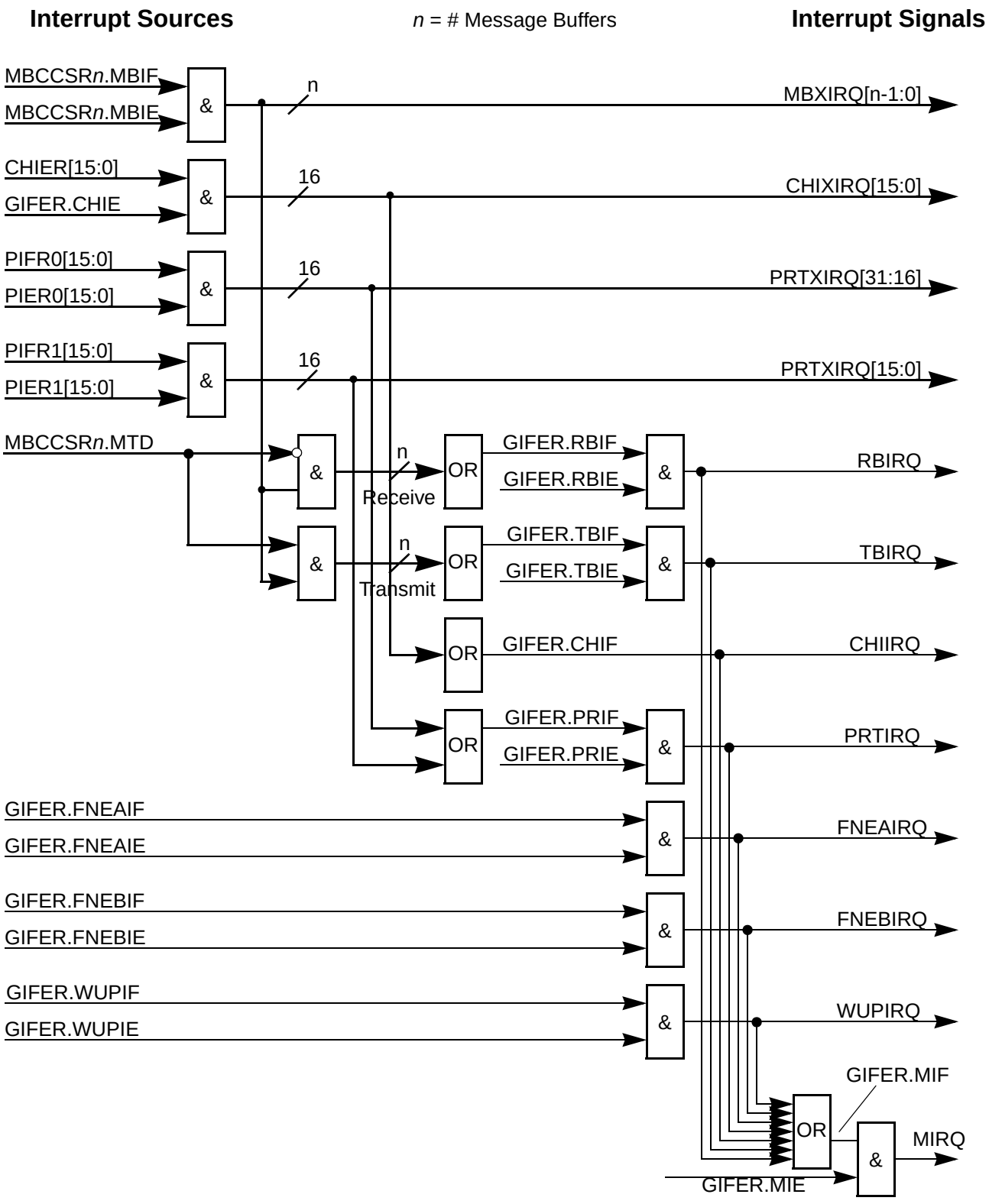


Figure 3-140. Scheme of cascaded interrupt request

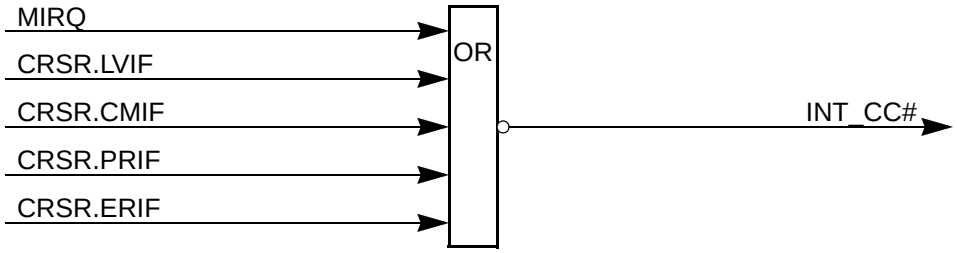


Figure 3-141. INT_CC# generation scheme

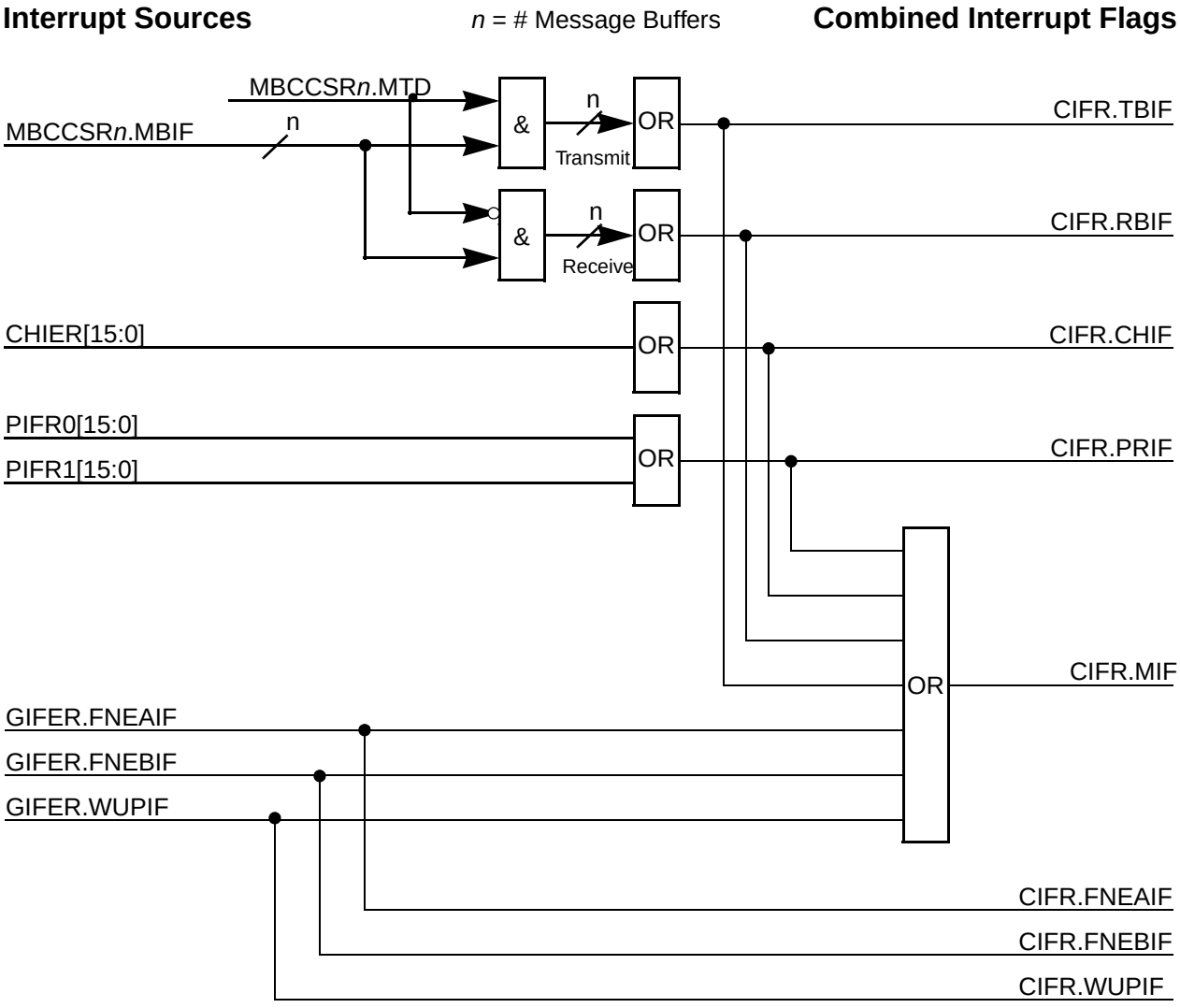


Figure 3-142. Scheme of combined interrupt flags

3.4.20 Clock Domain Crossing

The Clock Domain Crossing module CDC implements the signal crossing from the CHI clock domain to the PE clock domain and vice versa. The signal crossing logic is implemented as a three-stage pipe-line. Two pipe-line stages are used for clock synchronization; the third stage is used for pulse generation.

3.4.20.1 Clock Domain Crossing Signal Latency

Due to the clock domain crossing implementation, each signal from the PE to the CHI is delayed by at least two CHI clock cycles and by at most three CHI clock cycles. In terms of time, the signal latency time t_{lat} for a given CHI frequency f_{chi} is

$$\frac{2}{f_{chi}} \leq t_{lat} \leq \frac{3}{f_{chi}} \qquad \textit{Eqn. 3-26}$$

3.5 Lower FlexRay Bit Rate Support

The FlexRay module supports a number of lower bit rates on the FlexRay bus channels. The lower bit rates are implemented by modifying the duration of the microtick *pdMicrotick*, the number of samples per microtick *pSamplesPerMicrotick*, the number of samples per bit *cSamplesPerBit*, and the strobe offset *cStrobeOffset*. The application configures the FlexRay channel bit rate by setting the BITRATE field in the Module Configuration Register (MCR). The protocol values are set internally. The available bit rates, the related BITRATE field configuration settings and related protocol parameter values are shown in Table 3-110.

Table 3-110. FlexRay Channel Bit Rate Control

FlexRay Channel Bit Rate [Mbit/s]	MCR.BITRATE	*pdMicrotick* [ns]	*gdSampleClockPeriod* [ns]	*pSamplesPerMicrotick*	*cSamplesPerBit*	*cStrobeOffset*
10.0	000	25.0	12.5	2	8	5
8.0	011	25.0	12.5	2	10	6
5.0	001	25.0	25.0	1	8	5
2.5	010	50.0	50.0	1	8	5

NOTE

The bit rate of 8 Mbit/s is not defined by the *FlexRay Communications System Protocol Specification, Version 2.1 Rev A.*

3.6 Initialization Information

This section provides information for initializing and using the FlexRay module.

3.6.1 FlexRay Initialization Sequence

The full FlexRay module is reset with the hard reset. Additionally, the protocol engine is reset in the Stop Mode and as a result of the RESET protocol command issued using the Protocol Operation Control Register (POCR).

The hard reset resets all internal registers and all registers in the FlexRay module memory map. The protocol engine reset resets only the registers in the protocol engine. All registers in memory are not reset.

The following is an initialization sequence applicable to the FlexRay module after a hard reset

1. Configure FlexRay module
   - set the control bits in the Module Configuration Register (MCR)
2. Enable the FlexRay module
   - set the MEN bit in the Module Configuration Register (MCR)
   - the FlexRay module enters the Normal Mode
3. Configure the Protocol Engine
   - write the CONFIG command into the POCCMD field of the Protocol Operation Control Register (POCR)
   - write to the PCR0, ..., PCR30 registers to set all protocol parameters.
4. Configure the Message Buffers and FIFOs
   - set the number of message buffers used and the message buffer segmentation in the Message Buffer Segment Size and Utilization Register (MBSSUTR)
   - define the message buffer data size in the Message Buffer Data Size Register (MBDSR)
   - configure each message buffer by setting the configuration values in the Message Buffer Configuration, Control, Status Registers (MBCCSRn), Message Buffer Cycle Counter Filter Registers (MBCCFRn), Message Buffer Frame ID Registers (MBFIDRn), Message Buffer Index Registers (MBIDXRn)
   - configure the receive FIFOs
5. Start the FlexRay module as a FlexRay node
   - write the READY protocol command into the POCCMD field of the Protocol Operation Control Register (POCR)
   - now the FlexRay module enters the FlexRay protocol

After this sequence, the FlexRay module is configured as a FlexRay node and is ready to be integrated into the FlexRay cluster.

3.6.2 Number of Usable Message Buffers

This section describes the relationship between the number of message buffers that can be utilized and the required minimum CHI clock frequency.

The FlexRay module uses a sequential search algorithm to determine the individual message buffer assigned or subscribed to the next slot. This search must be finished within one FlexRay slot. The shortest FlexRay slot is an empty dynamic slot. An empty dynamic slot is a minislot and consists of *gdMinislot*

macroticks with a nominal duration of *gdMacrotick*. The minimum duration of a corrected macrotick is $gdMacrotick_{min}$ equaling 39 µµT. This results in a minimum slot length of:

$$\Delta_{slotmin} = 39 \cdot pdMicrotick \cdot gdMinislot \quad \textit{Eqn. 3-27}$$

The search engine is located in the CHI and runs on the CHI clock. It evaluates one individual message buffer per CHI clock cycle. For internal status update and double buffer commit operations, and as a result of the clock domain crossing jitter, an additional amount of 10 CHI clock cycles is required to ensure correct operation. For a given number of message buffers and for a given CHI clock frequency f_{chi}, this results in a search duration of

$$\Delta_{search} = \frac{1}{f_{chi}} \cdot (\#\ \text{MessageBuffers} + 10) \quad \textit{Eqn. 3-28}$$

The message buffer search must be finished within one slot which requires that Equation 3-29 must be fulfilled

$$\Delta_{search} \leq \Delta_{slotmin} \quad \textit{Eqn. 3-29}$$

This results in the formula given in Equation 3-30 which determines the required minimum CHI frequency for a given number of message buffers that are utilized.

$$f_{chi} \geq \frac{\#\ \text{MessageBuffers} + 10}{39 \cdot pdMicrotick \cdot gdMinislot} \quad \textit{Eqn. 3-30}$$

The minimum CHI frequency for a selected set of relevant protocol parameters is given in Table 3-111.

Table 3-111. Minimum f_{chi} [MHz] examples (128 message buffers)

pdMicrotick [ns]	*gdMinislot*					
	2	3	4	5	6	7
25.0	70.77	47.18	35.39	28.31	23.59	20.22
50.0	35.39	23.59	17.70	14.16	11.80	10.11

3.7 Application Information

3.7.1 Shut Down Sequence

This section describes a safe shut down sequence to stop the FlexRay module gracefully. The main targets of this sequence are

- do not send invalid data on the FlexRay bus
- do not corrupt FlexRay bus anddo not disturb ongoing communication
- finish all ongoing reception

Firstly, the application must disable all message buffers by triggering the EDT trigger bit in the Message Buffer Configuration, Control, Status Registers (MBCCSRn), until the EDS flag is cleared by the FlexRay module. This ensures that no transmission is started by the FlexRay module. If all message buffers are disabled, the application issues the HALT command to the PE using the Protocol Operation Control Register (POCR). The PE then waits for the end of the communication cycle and goes into the *POC:halt*

state. The application can observe this state change in the PROTSTATE field of the Protocol Status Register 0 (PSR0).

3.7.2 Protocol Control Command Execution

This section considers the issues of the protocol control command execution.

The application issues any of the protocol control commands listed in the POCCMD field of Table 3-15 by writing the command to the POCCMD field of the Protocol Operation Control Register (POCR). As a result the FlexRay module sets the BSY bit while the command is transferred to the PE. When the PE has accepted the command, the BSY flag is cleared. All commands are accepted by the PE.

The PE maintains a protocol command vector. For each command that was accepted by the PE, the PE sets the corresponding command bit in the protocol command vector. If a command is issued while the corresponding command bit is set, the command is not queued and is lost.

If the command execution block of the PE is idle, it selects the next accepted protocol command with the highest priority from the current protocol command vector according to the protocol control command priorities given in Table 3-112. If the current protocol state does not allow the execution of this protocol command (see POC state changes in *FlexRay Communications System Protocol Specification, Version 2.1 Rev A*) the FlexRay module asserts the illegal protocol command interrupt flag IPC_IF in the Protocol Interrupt Flag Register 1 (PIFR1). The protocol command is not executed in this case.

Some protocol commands may be interrupted by other commands or the detection of a fatal protocol error as indicated by Table 3-112. If the application issues the RESET, FREEZE, or READY command, or if the the PE detects a fatal protocol error, some commands already stored in the command vector are removed from this vector.

Table 3-112. Protocol Control Command Priorities

Protocol Command	Priority	Interrupted By	Cleared and Terminated By
RESET	(highest) 1	none	
FREEZE	2		RESET
READY	3		RESET
CONFIG_COMPLETE	3		RESET

Table 3-112. Protocol Control Command Priorities

Protocol Command	Priority	Interrupted By	Cleared and Terminated By
ALL_SLOTS	4	RESET, FREEZE, READY, CONFIG_COMPLET, fatal protocol error	RESET, FREEZE, READY, CONFIG_COMPLETE, fatal protocol error
ALLOW_COLDSTART	5		RESET
RUN	6		RESET, FREEZE, fatal protocol error
WAKEUP	7		RESET, FREEZE, fatal protocol error
DEFAULT_CONFIG	8		RESET, FREEZE, fatal protocol error
CONFIG	9		RESET
HALT	(lowest) 10		RESET, FREEZE, READY, CONFIG_COMPLETE, fatal protocol error

3.7.3 Protocol Reset Command

The section considers the issues of the protocol RESET command.

The application issues the protocol reset command by writing the RESET command code to the POCCMD field of the Protocol Operation Control Register (POCR). As a result, the PE stops its operation immediately, the FlexRay bus ports put into their idle state, and no more data or status information is sent to the CHI. The lack of PE signals stops all message buffer operations in the CHI. In particular, the message buffers that are currently under internal use remain internally locked. To overcome this message buffer internal lock situation, the application must put the protocol into the *POC:default config* state. This releases all internal message buffer locks.

The following sequence must be executed by the application to put the protocol into the *POC:default config* state.

1. Repeatedly send Protocol Command FREEZE via Protocol Operation Control Register (POCR), until the freeze bit FRZ in Protocol Status Register 1 (PSR1) is set.
2. Repeatedly send Protocol Command DEFAULT_CONFIG via Protocol Operation Control Register (POCR), until the freeze bit FRZ bit in Protocol Status Register 1 (PSR1) is cleared and the PROTSTATE field in Protocol Status Register 0 (PSR0) is set to DEFAULT_CONFIG.

Chapter 4
Port Integration Module (PIM)

4.1 Introduction

4.1.1 Overview

The Port Integration Module implements the interfaces between the FlexRay IP block, the peripheral modules, and the I/O pins.

4.1.2 Features

The Port Integration Module includes these distinctive features:

- Pad control for all functional pads including:
  - drive strength enable (DSE), via a control register
  - pull enable (PUE), via a control register
  - pull select (PUS), via a control register
- Pin multiplexing and direction control for reset mode

4.1.3 Modes of Operation

The Port Integration Module can be put into the following modes:

- Functional Mode

  In this mode, the module drives each associated pin and has complete control of the direction of that pin. The drive strength and pullup/pulldown enable are controlled via a set of control registers.
- Reset Mode

  In this mode, the pin configuration is changed for:
  - clock output control: CLK_S0 and CLK_S1
  - host interface control: IF_SEL0 and IF_SEL1

  The control signals become available on the corresponding pins in reset mode. Refer to Chapter 6, “Clocks and Reset Generator (CRG)” for reset mode details.

This is a high level description only; detailed descriptions of operating modes are contained in later sections.

4.2 External Signal Description

For detailed descriptions of particular pins and signals, refer to Section 2.4, “Signal Descriptions”.

4.2.1 Functional Mode

Table 4-1. Pin Functions (Functional Mode)

Name	Function	Direction	Special Configuration[1]
Host Interface			
A[6:1]/XADDR[14:19]	AMI/MPC address bus; HCS12 expanded address lines. A1 is the LSB of the AMI/MPC address bus; XADDR14 is the LSB of the HCS12 expanded address lines	Input	PU/PD
A[7:9]	AMI/MPC address bus	Input	PU/PD
OE#/ACS0	AMI/MPC read output enable signal; HCS12 address select input	Input	PU/PD
A[12:11]/ACS[2:1]	AMI/MPC address bus; HCS12 address select inputs	Input	PU/PD
BSEL[1:0]#/DBG[0:1]	AMI/MPC byte select; Debug strobe point	Input/Output	DC/PU/PD
D[15:8]/PB[0:7]	AMI/MPC data bus; HCS12 multiplexed address/data bus. D15 is the MSB of the AMI/MPC data bus; PB0 is the LSB of the HCS12 address/data bus	Input/Output	DC/PU/PD
D[7:0]/PA[0:7]	AMI/MPC data bus; HCS12 multiplexed address/data bus. D0 is the LSB of the AMI/MPC data bus; PA7 is the MSB of the HCS12 address/data bus	Input/Output	DC/PU/PD
CE#/LSTRB	AMI/MPC chip select signal; HCS12 low-byte strobe signal	Input	PU/PD
WE#/RW_CC#	AMI write enable signal; HCS12 read/write select signal	Input	PU/PD
A10/ECLK_CC	AMI/MPC address bus/; HCS12 clock input	Input	PU/PD
Physical Layer Interface			
RXD_BG[2:1]	PHY Data receiver input	Input	PU/PD
TXEN[2:1]#	Transmit enable for PHY	Output	DC
TXD_BG[1:2]/IF_SEL[1:0]	PHY Data transmitter output / Host interface select	Input/Output	DC
Clock Interface			
CHICLK_CC	External CHI clock input selectable	Input	-
CLKOUT	Controller clock output selectable between disabled, 4/10/40 MHz	Output	DC
Other			
RESET#	Hardware reset input	Input	PD
INT_CC#	Controller interrupt output	Output	DC/OD
TEST	Factory Test mode select — must be tied to logic low in application	Input	PD

Table 4-1. Pin Functions (Functional Mode) (continued)

Name	Function	Direction	Special Configuration[1]
DBG[3:2]/CLK_S[1:0]	Debug strobe point / External CHI clock input select	Output	DC
Oscillator			
EXTAL/CC_CLK	Crystal driver / External clock	Input	-
XTAL	Crystal driver	Input	-

1 Acronyms:
PC – (Pullup/pulldown Controlled) Register controlled internal weak pullup/pulldown for a pin in the input mode
PU/PD – (Pullup/pulldown) Internal weak pullup/pulldown for a pin in the input mode
DC – (Drive strength Controlled) Register controlled drive strength for a pin in the output mode

The TEST and RESET pads have internal pulldown resistors, where the pulldown resistor on RESET is disabled when TEST is asserted. If the RESET pad gets disconnected, the device goes into a safe state (reset mode).

4.2.2 Reset Mode

This pin configuration is enabled in reset mode only. Refer to Chapter 6, “Clocks and Reset Generator (CRG)” for reset mode details. When the device is in reset mode, the corresponding pads go into input mode with pull enabled.

Table 4-2. Pin Functions (Reset Mode)

Name	Direction	Special Configuration[1]
TXD_BG2/IF_SEL0	Input	PU
TXD_BG1/IF_SEL1	Input	PD
DBG[3:2]/CLK_S[1:0]	Input	PD

1 Acronyms:
PU/PD – (Pullup/pulldown) Internal weak pullup/pulldown for a pin in the input mode

4.3 PIM Memory Map and Registers

This section provides a detailed description of all registers in the Port Integration Module.

Only 16-bit reads and 8-bit and 16-bit writes are allowed to all registers.

Table 4-3. Port Integration Module Memory Map

Address	Use	Access
0x00F0	Part ID Register (PIDR)	R
0x00F2	ASIC Version Number Register (AVNR)	R
0x00F4	Host Interface Pins Drive Strength Register (HIPDSR)	R/W
0x00F6	Physical Layer Pins Drive Strength Register (PLPDSR)	R/W
0x00F8	Host Interface Pins Pullup/pulldown Enable Register (HIPPER)	R/W

Table 4-3. Port Integration Module Memory Map

Address	Use	Access
0x00FA	Host Interface Pins Pullup/pulldown Control Register (HIPPCR)	R/W
0x00FC	Physical Layer Pins Pullup/pulldown Enable Register (PLPPER)	R/W
0x00FE	Physical Layer Pins Pullup/pulldown Control Register (PLPPCR)	R/W

4.3.1 Port Integration Module Registers

4.3.1.1 Part ID Register (PIDR)

Address in MFR4310 = 0x00F0

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0
W																
Reset	0	1	0	0	0	0	1	1	0	0	0	1	0	0	0	0

Figure 4-1. Part ID Register (PIDR)

This register provides the part ID (4310) in binary coded decimal.

4.3.1.2 ASIC Version Number Register (AVNR)

Address in MFR4310 = 0x00F2

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 4-2. ASIC Version Number Register (AVNR) (for Maskset 1M63J)

This register provides the ASIC version number in binary coded decimal. The content of this register is dependent on the maskset number. Figure 4-2 shows the content and reset values for maskset 1M63J. See Section 2.3.2, “Part ID and Module Version Number Assignments” for details of maskset numbers.

4.3.1.3 Host Interface Pins Drive Strength Register (HIPDSR)

Address in MFR4310 = 0x00F4 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0				
W													CLKOUT	DBG[3:2]	INT_CC#	D[0:15]/ PA[0:7]/PB[0:7]
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figure 4-3. Host Interface Pins Drive Strength Register (HIPDSR)

This register controls the drive strength of the host interface, interrupt, debug, and output clock pins.

Table 4-4. HIPDSR Field Descriptions

Field	Description
0 D[0:15]/ PA[0:7]/PB[0:7]	Host interface output data drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
1 INT_CC#	Interrupt output drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
2 DBG[3:2]	Debug output (bits 3 and 2 only) drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
3 CLKOUT	Output clock drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full

4.3.1.4 Physical Layer Pins Drive Strength Register (PLPDSR)

Address in MFR4310 = 0x00F6 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	TXD_BG2	TXD_BG1	TXEN2#	TXEN1#
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Figure 4-4. Physical Layer Pins Drive Strength Register (PLPDSR)

This register controls the drive strength of the FlexRay physical layer pins.

Table 4-5. PLPDSR Field Descriptions

Field	Description
0 TXEN1#	Transmit enable (channel A) output drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
1 TXEN2#	Transmit enable (channel B) output drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
2 TXD_BG1	Transmit data (channel A) output drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full
3 TXD_BG2	Transmit data (channel B) output drive strength control 0 Pin drive strength is reduced to 1/3 of full strength 1 Pin drive strength is full

4.3.1.5 Host Interface Pins Pullup/pulldown Enable Register (HIPPER)

Address in MFR4310 = 0x00F8 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	WE#/RW_CC#	CE#/LSTRB	D[0:15]/ PA[0:7]/PB[0:7]	A12/ACS2	A11/ACS1	OE#/ACS0	BSEL[1:0]#	A[10:7]	A6/XADDR14	A5/XADDR15	A4/XADDR16	A3/XADDR17	A2/XADDR18	A1/XADDR19
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-5. Host Interface Pins Pullup/pulldown Enable Register (HIPPER)

This register enables/disables the pullups/pulldowns of the host interface pins.

Table 4-6. HIPPER Field Descriptions

Field	Description
0 A1/XADDR19	AMI/MPC address bit 1 / HCS12 expanded address bit 19 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
1 A2/XADDR18	AMI/MPC address bit 2 / HCS12 expanded address bit 18 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
2 A3/XADDR17	AMI/MPC address bit 3 / HCS12 expanded address bit 17 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
3 A4/XADDR16	AMI/MPC address bit 4 / HCS12 expanded address bit 16 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
4 A5/XADDR15	AMI/MPC address bit 5 / HCS12 expanded address bit 15 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
5 A6/XADDR14	AMI/MPC address bit 6 / HCS12 expanded address bit 14 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
6 A[10:7]	AMI/MPC address bits 7 through 10 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
7 BSEL[1:0]#	AMI/MPC byte select pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
8 OE#/ACS0	AMI/MPC output enable / HCS12 address select bit 0 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
9 A11/ACS1	AMI/MPC address bit 11 / HCS12 address select bit 1 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled

Table 4-6. HIPPER Field Descriptions (continued)

Field	Description
10 A12/ACS2	AMI/MPC address bit 12 / HCS12 address select bit 2 pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
11 D[0:15]/ PA[0:7]/PB[0:7]	Host interface input data pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
12 CE#/LSTRB	AMI/MPC chip enable / HCS12 low-byte strobe pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
13 WE#/RW_CC#	AMI write enable / HCS12 read/write select pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled

4.3.1.6 Host Interface Pins Pullup/pulldown Control Register (HIPPCR)

Address in MFR4310 = 0x00FA — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	WE#/RW_CC#	CE#/LSTRB	D[0:15]/ PA[0:7]/PB[0:7]	A12/ACS2	A11/ACS1	OE#/ACS0	BSEL[1:0]#	A[10:7]	A6/XADDR14	A5/XADDR15	A4/XADDR16	A3/XADDR17	A2/XADDR18	A1/XADDR19
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-6. Host Interface Pins Pullup/pulldown Control Register (HIPPCR)

This register enables/disables the pullups/pulldowns of the host interface pins.

Table 4-7. HIPPCR Field Descriptions

Field	Description
0 A1/XADDR19	AMI/MPC address bit 1 / HCS12 expanded address bit 19 pullup/pulldown control 0 pulldown 1 pullup
1 A2/XADDR18	AMI/MPC address bit 2 / HCS12 expanded address bit 18 pullup/pulldown control 0 pulldown 1 pullup
2 A3/XADDR17	AMI/MPC address bit 3 / HCS12 expanded address bit 17 pullup/pulldown control 0 pulldown 1 pullup
3 A4/XADDR16	AMI/MPC address bit 4 / HCS12 expanded address bit 16 pullup/pulldown control 0 pulldown 1 pullup
4 A5/XADDR15	AMI/MPC address bit 5 / HCS12 expanded address bit 15 pullup/pulldown control 0 pulldown 1 pullup

Table 4-7. HIPPCR Field Descriptions (continued)

Field	Description
5 A6/XADDR14	AMI/MPC address bit 6 / HCS12 expanded address bit 14 pullup/pulldown control 0 pulldown 1 pullup
6 A[10:7]	AMI/MPC address bits 7 through 10 pullup/pulldown control 0 pulldown 1 pullup
7 BSEL[1:0]#	AMI/MPC byte select pullup/pulldown control 0 pulldown 1 pullup
8 OE#/ACS0	AMI/MPC output enable / HCS12 address select bit 0 pullup/pulldown control 0 pulldown 1 pullup
9 A11/ACS1	AMI/MPC address bit 11 / HCS12 address select bit 1 pullup/pulldown control 0 pulldown 1 pullup
10 A12/ACS2	AMI/MPC address bit 12 / HCS12 address select bit 2 pullup/pulldown control 0 pulldown 1 pullup
11 D[0:15]/ PA[0:7]/PB[0:7]	Host interface input data pullup/pulldown control 0 pulldown 1 pullup
12 CE#/LSTRB	AMI/MPC chip enable / HCS12 low-byte strobe pullup/pulldown control 0 pulldown 1 pullup
13 WE#/RW_CC#	AMI write enable / HCS12 read/write select pullup/pulldown control 0 pulldown 1 pullup

4.3.1.7 Physical Layer Pins Pullup/pulldown Enable Register (PLPPER)

Address in MFR4310 = 0x00FC — Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RXD_BG2	RXD_BG1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-7. Physical Layer Pins Pullup/pulldown Enable Register (PLPPER)

This register enables/disables the pullups/pulldowns of the FlexRay physical layer pins.

Table 4-8. PLPPER Field Descriptions

Field	Description
0 RXD_BG1	Receive data (channel A) pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled
1 RXD_BG2	Receive data (channel B) pullup/pulldown enable 0 pullup/pulldown disabled 1 pullup/pulldown enabled

4.3.1.8 Physical Layer Pins Pullup/pulldown Control Register (PLPPCR)

Address in MFR4310 = 0x00FE Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	RXD_BG2	RXD_BG1
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4-8. Physical Layer Pins Pullup/pulldown Control Register (PLPPCR)

This register enables/disables the pullups/pulldowns of the host interface pins.

Table 4-9. PLPPCR Field Descriptions

Field	Description
0 RXD_BG1	Receive data (channel A) pullup/pulldown control 0 pulldown 1 pullup
1 RXD_BG2	Receive data (channel B) pullup/pulldown control 0 pulldown 1 pullup

4.4 Functional Description

The Port Integration Module provides the capability to configure all functional I/O pins regarding:

- Output drive with two selectable drive strengths
- Pullup or pulldown
- Pin multiplexing and pin configuration constraints for reset mode

4.4.1 Functional Mode

In functional mode, the Port Integration Module controls the functional interface:

- Host Interface
- Physical Layer Interface
- Clock Interface

The module provides pullup/pulldown and drive strength control through configuration registers via the IPBus interface. The actual control registers are described in Section 4.3, “PIM Memory Map and Registers”.

4.4.2 Reset Mode

In reset mode, the Port Integration Module provides access to four configuration pins for clock output control in the CRG and external bus interface (EBI) control in the FlexRay IP block. In this case the corresponding pins are forced to input mode with pull enabled.

Table 4-10. Reset Mode Interface

Name	Direction	Special Configuration
TXD_BG2/IF_SEL0	Input	PU
TXD_BG1/IF_SEL1	Input	PD
DBG[3:2]/CLK_S[1:0]	Input	PD

See Section 4.2.2, “Reset Mode” and Chapter 6, “Clocks and Reset Generator (CRG)” for reset mode details.

Chapter 5
Dual Output Voltage Regulator (VREG3V3V2)

5.1 Introduction

The VREG3V3V2 is a dual output voltage regulator providing two separate 2.5 V (typical) supplies differing in the amount of current that can be sourced. The regulator input voltage range is from 3.3 V up to 5 V (typical).

5.1.1 Features

The block VREG3V3V2 includes these distinctive features:

- Two parallel, linear voltage regulators
  - Bandgap reference
- Power-on reset (POR)
- Low-voltage reset (LVR)

5.1.2 Modes of Operation

VREG3V3V2 can operate in two modes on MFR4310:

- Full-performance mode (FPM)

  The regulator is active, providing the nominal supply voltage of 2.5 V with full current sourcing capability at both outputs. Features LVR (low-voltage reset) and POR (power-on reset) are available.
- Shutdown mode

  Controlled by V_{DDR}.

  This mode is characterized by minimum power consumption. The regulator outputs are in a high impedance state; only the POR feature is available, and LVR is disabled.

  This mode must be used to disable the chip internal regulator VREG3V3V2, i.e., to bypass the VREG3V3V2 to use external supplies.

5.1.3 Block Diagram

Figure 5-1 shows the function principle of VREG3V3V2 by means of a block diagram. The regulator core REG consists of two parallel sub-blocks, REG1 and REG2, providing two independent output voltages.

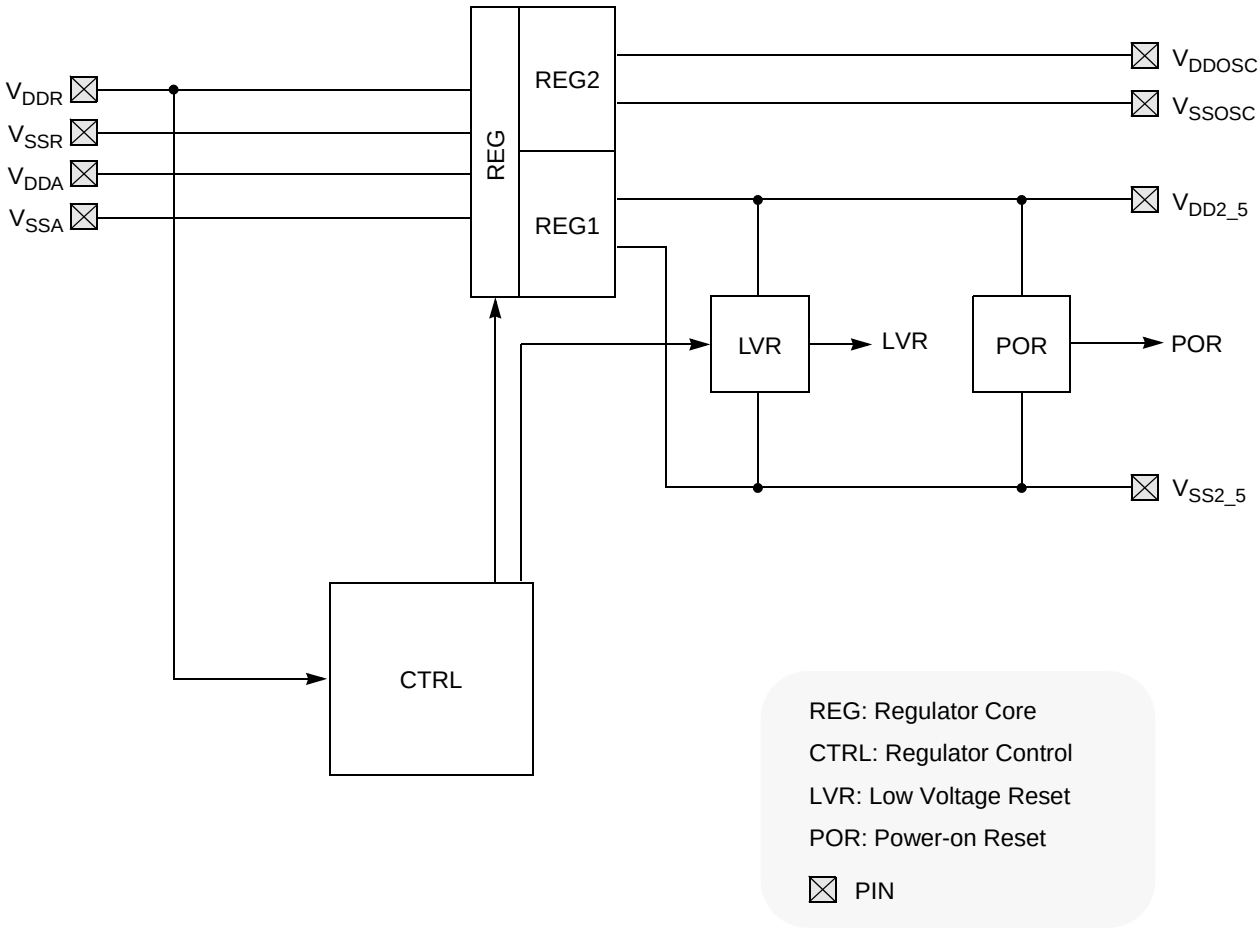


Figure 5-1. VREG3V3 Block Diagram

5.2 External Signal Description

Due to the nature of VREG3V3V2 being a voltage regulator providing the chip internal power supply voltages most signals are power supply signals connected to pads.

Table 5-1 shows all signals of VREG3V3V2 associated with pins.

Table 5-1. VREG3V3V2 — Signal Properties

Name	Port	Function	Reset State	Pullup
V_{DDR}	—	VREG3V3V2 power input (positive supply)	—	—
V_{SSR}	—	VREG3V3V2 power input (ground)	—	—
V_{DDA}	—	VREG3V3V2 quiet input (positive supply)	—	—
V_{SSA}	—	VREG3V3V2 quiet input (ground)	—	—
V_{DD2_5}	—	VREG3V3V2 primary output (positive supply)	—	—
V_{SS2_5}	—	VREG3V3V2 primary output (ground)	—	—
V_{DDOSC}	—	VREG3V3V2 secondary output (positive supply)	—	—
V_{SSOSC}	—	VREG3V3V2 secondary output (ground)	—	—

NOTE

Check device overview chapter for connectivity of the signals.

5.2.1 V_{DDR}, V_{SSR} — Regulator Power Input

Signal V_{DDR} is the power input of VREG3V3V2. All currents sourced into the regulator loads flow through this pin. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between V_{DDR} and V_{SSR} can smoothen ripple on V_{DDR}.

For entering shutdown mode, pin V_{DDR} must be tied to ground. In that case, V_{DD2_5}/V_{SS2_5} and V_{DDOSC}/V_{SSOSC} must be provided externally.

5.2.2 V_{DDA}, V_{SSA} — Regulator Reference Supply

Signals V_{DDA}/V_{SSA} which are supposed to be relatively quiet are used to supply the analog parts of the regulator. Internal precision reference circuits are supplied from these signals. A chip external decoupling capacitor (100 nF...220 nF, X7R ceramic) between V_{DDA} and V_{SSA} can further improve the quality of this supply.

5.2.3 V_{DD2_5}, V_{SS2_5} — Regulator Output1 (Core Logic)

Signals V_{DD2_5}/V_{SS2_5} are the primary outputs of VREG3V3V2 that provide the power supply for the core logic. These signals are connected to device pins to allow external decoupling capacitors (100 nF...220 nF, X7R ceramic).

In shutdown mode an external supply at V_{DD2_5}/V_{SS2_5} can replace the voltage regulator.

5.2.4 V_{DDOSC}, V_{SSOSC} — Regulator Output2 (OSC)

Signals V_{DDOSC}/V_{SSOSC} are the secondary outputs of VREG3V3V2 that provide the power supply for the oscillator. These signals are connected to device pins to allow external decoupling capacitors (100 nF...220 nF, X7R ceramic).

In shutdown mode an external supply at V_{DDOSC}/V_{SSOSC} can replace the voltage regulator.

5.3 Functional Description

Block VREG3V3V2 is a voltage regulator as depicted in Figure 5-1. The regulator functional elements are the regulator core (REG), a power-on reset module (POR) and a low-voltage reset module (LVR). There is also the regulator control block (CTRL) which manages the operating modes of VREG3V3V2.

5.3.1 REG — Regulator Core

VREG3V3V2, respectively its regulator core has two parallel, independent regulation loops (REG1 and REG2) that differ only in the amount of current that can be sourced to the connected loads. Therefore, only REG1 providing the supply at V_{DD2_5}/V_{SS2_5} is explained. The principle is also valid for REG2.

The regulator is a linear series regulator with a bandgap reference in its full-performance mode and a voltage clamp in reduced-power mode. All load currents flow from input V_{DDR} to V_{SS2_5} or V_{SSOSC}, the reference circuits are connected to V_{DDA} and V_{SSA}.

5.3.2 Full-performance Mode

In full-performance mode, a fraction of the output voltage (V_{DD2_5}) and the bandgap reference voltage are fed to an operational amplifier. The amplified input voltage difference controls the gate of an output driver.

5.3.3 POR — Power On Reset

This functional block monitors output V_{DD2_5}. If V_{DD2_5} is below V_{PORD}, signal POR is high; if it exceeds V_{PORD}, the signal goes low. The transition to low forces the CPU into the power-on sequence.

Due to its role during chip power-up, this module must be active in all operating modes of VREG3V3V2.

5.3.4 LVR — Low Voltage Reset

Block LVR monitors the primary output voltage V_{DD2_5}. If it drops below the assertion level (V_{LVRA}) signal LVR asserts and when rising above the deassertion level (V_{LVRD}) signal LVR deasserts again. The LVR function is available only in full-performance mode.

5.3.5 CTRL — Regulator Control

This part contains digital functionality needed to control the operating modes.

5.4 Resets

This subsection describes how VREG3V3V2 controls the reset of the CC. The reset values of registers and signals are provided in Section 3.3, “Memory Map and Register Description”. Possible reset sources are listed in Table 5-2.

Table 5-2. VREG3V3V2 — Reset Sources

Reset Source	Local Enable
Power-on reset	Always active
Low-voltage reset	Always active

5.4.1 Power On Reset

During chip power-up the digital core may not work if its supply voltage V_{DD2_5} is below the POR deassertion level (V_{PORD}). Therefore, signal POR, which forces the other blocks of the device into reset, is kept high until V_{DD2_5} exceeds V_{PORD}. Then POR becomes low and the reset generator of the device continues the start-up sequence.

5.4.2 Low Voltage Reset

For information on low-voltage reset see Section 5.3.4, “LVR — Low Voltage Reset”.

Chapter 6
Clocks and Reset Generator (CRG)

6.1 Introduction

6.1.1 Overview

This document describes the CRG operation in functional mode and only those aspects of it which are useful users. Additional topics as system clock generation or functionality while the CRG is in another operational modes are out of the scope of this documentation.

6.1.2 Features

The CRG includes the following main features:

- System reset generation from power-on and external reset events
- System reset generation from low voltage reset event
- Controllable system reset generation under low quality clock situations (clock monitor)
- System reset indication
- Host interface selection
- Control signals selection for *CLKOUT* clock output
- System clocks generation
- Reset glitch filter

6.2 MFR4310 Relevant Pins for the CRG

Table 6-1 describes the MFR4310 pins relevant for the CRG block.

Table 6-1. MFR4310 **Relevant Pins for the CRG**

Pin Name[1]	In/Out	Pin type[2,3,4]	Functional Description
TXD_BG2/IF_SEL0	I/O	DC/PU	PHY Data transmitter output / Host interface select
TXD_BG1/IF_SEL1	I/O	DC/PD	PHY Data transmitter output / Host interface select
CHICLK_CC	I	-	External CHI clock input – *selectable*
CLKOUT	I/O	DC	Controller clock output–selectable from: disabled, 4/10/40 MHz
RESET#	I	PD	Hardware reset input
INT_CC#	O	OD/DC	Controller interrupt output
TEST	I	PD	Factory Test mode select– *should be tied to logic low in application*
DBG[3:2]/CLK_S[1:0]	I/O	DC/PD	Debug strobe point / Output clock select
EXTAL/CLK_CC	I	-	Crystal driver / External clock pin
XTAL	I	-	Crystal driver pin

1 # – signal is active-low

2 Acronyms:
PC **–** (Pullup/pulldown Controlled) Register controlled internal weak pullup/pulldown for a pin in the input mode
PU/PD – (Pullup/Pulldown) Internal weak pullup/pulldown for a pin in the input mode
DC – (Drive strength Controlled) Register controlled drive strength for a pin in the output mode
OD – (Open Drain) Output pin with open drain
Z – Tristated pin

3 No load allowed except for bypass capacitors.

4 Reset state: All pins with the PC option – pullup/pulldown is disabled,
All pins with the DC option – have full drive strength

6.3 CRG Registers

The bits in the CRG registers are set by the CRG synchronous to the CHI clock signal. The *system reset* signal is a hard reset for CRG registers.

6.3.1 Detection Enable Register (DER)

Address in MFR4310 = 0x00E0 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CMIE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6-1. Detection Enable Register (DER)

Table 6-2. DER Field Descriptions

Field	Description
0 CMIE	Clock Monitor Mechanism Enable 0 Clock monitor mechanism disabled 1 Clock monitor mechanism enabled

NOTE

After reset, the clock monitor mechanism is disabled.

6.3.2 Clock and Reset Status Register (CRSR)

Address in MFR4310 = 0x00E2 Write: Any Time

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	CDCV		ECS	0	0	0	0	ERIF	PRIF	CMIF	LVIF
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	–	–	–	–

Figure 6-2. Clock and Reset Status Register (CRSR)

Table 6-3. CRSR Field Descriptions

Field	Description
0 LVIF	Low Voltage Reset Interrupt Flag — set when a low-voltage reset has occurred. Cleared by writing a 1. Writing 0 has no effect.
1 CMIF	Clock Monitor Reset Interrupt Flag — set when a clock monitor reset has occurred. Cleared by writing a 1. Writing 0 has no effect. **Note:** If LVIF bit or PRIF bit is set to 1 then the CMIF bit value is 0.
2 PRIF	Power-on Reset Interrupt Flag — set when a power-on reset has occurred. Cleared by writing a 1. Writing 0 has no effect.
3 ERIF	External Reset Interrupt Flag — set when a external reset has occurred. Cleared by writing a 1. Writing 0 has no effect. **Note:** If LVIF bit or PRIF bit is set to 1 then the ERIF bit value is 0.
8 ECS	CHI Clock Source 0 CHI blocks are clocked by EXTAL 1 CHI blocks are clocked by CHICLK_CC
10-9 CDCV	CLKOUT Division Control Value — contains sampled value of CLK_S[1:0]. The CRG writes this value after a power-on, low-voltage or clock monitor reset, according to the values sampled on the CLK_S[1:0] pins. See Table 2-5 for coding.

NOTE

On a power-on or low-voltage reset, CMIF and PRIF are both cleared to 0.

6.4 Functional Description

6.4.1 Reset Generation

The CRG provides a system reset in any of the following events: power-on, low-voltage, or clock monitor failure (if enabled) detected, low level detected at the *RESET#* pin. Entry into reset is asynchronous and does not require a clock. However, the MFR4310 cannot sequence out of reset in Functional Mode without a system clock. Table 6-4 depicts reset sources priorities.

The CRG scans, during different periods depending on the origin of the reset source, the interface type, the AMI clock source and the *CLKOUT* mode selection pins: *IF_SEL[1:0]* and *CLK_S[1:0]*.

Table 6-4. CRG Reset Sources Priorities

Reset Source	Block to Reset	Priority
Power-on Reset	Whole device	High
Low voltage or Clock Monitor Failure (if enabled) Reset	Whole device	
External Reset	Whole device	Low

NOTE

After the CRG has started a reset procedure, it does not abandon it unless a reset event with more priority is detected. A reset procedure with the same priority as the currently running one stops the previous procedure and is executed.

6.4.1.1 Power-on Reset

When the power-on reset signal is asserted the CRG asserts the system reset signal. The CRG synchronously deasserts the system reset signal approximately 16420 *EXTAL/CLK_CC* clock periods after the deassertion of the power-on reset signal.

The CRG resets the DER.CMIE bit to 0, asserts the *INT_CC#* interrupt line, and sets the power-on reset interrupt flag, CRSR.PRIF, on the rising edge of the power-on reset signal.

NOTE

The CRG deasserts the INT_CC# signal when the CRSR.PRIF, CRSR.LVIF, CRSR.CMIF and CRSR.ERIF bits are 0.

Figure 6-3 illustrates the power-on reset situation.

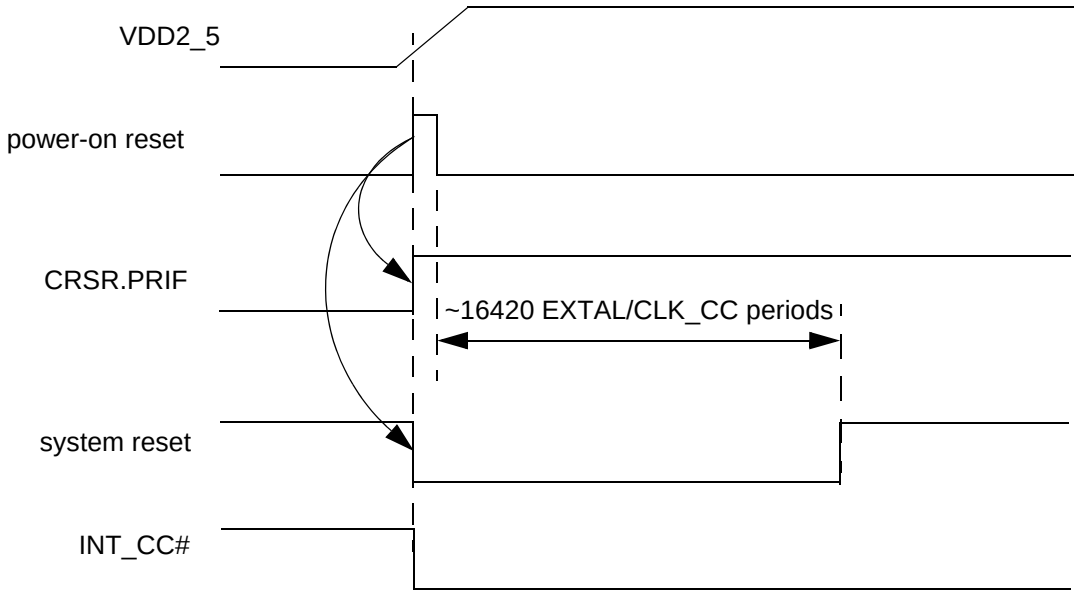


Figure 6-3. CRG Power On Reset

6.4.1.2 Low Voltage and Clock Monitor Reset

When the low voltage reset or clock monitor failure (if enabled) signal is asserted, the CRG asserts the system reset signal. The CRG synchronously deasserts the system reset signal approximately 16420 *EXTAL/CLK_CC* clock periods after the deassertion of the low voltage reset or clock monitor failure signal.

The CRG resets the DER.CMIE bit to 0, asserts the *INT_CC#* interrupt line, and sets the low voltage reset interrupt flag, CRSR.LVIF, on the rising edge of the low voltage reset signal.

The CRG resets the DER.CMIE bit to 0, asserts the *INT_CC#* interrupt line, and sets the clock monitor failure interrupt flag, *CRSR.CMIF*, on the rising edge of the clock monitor failure signal (if this is enabled).

NOTE

The CRG deasserts the INT_CC# signal when the CRSR.PRIF, CRSR.LVIF, CRSR.CMIF and CRSR.ERIF bits are 0.

Figure 6-4 and Figure 6-5 show the operations performed by the CRG when a low voltage reset or a clock monitor failure (if enabled) signal occurs.

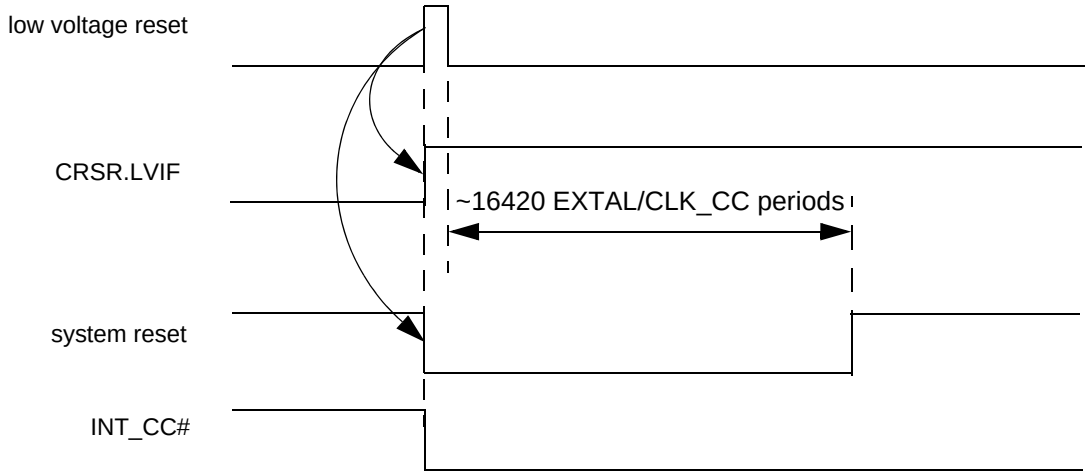


Figure 6-4. Low Voltage Reset

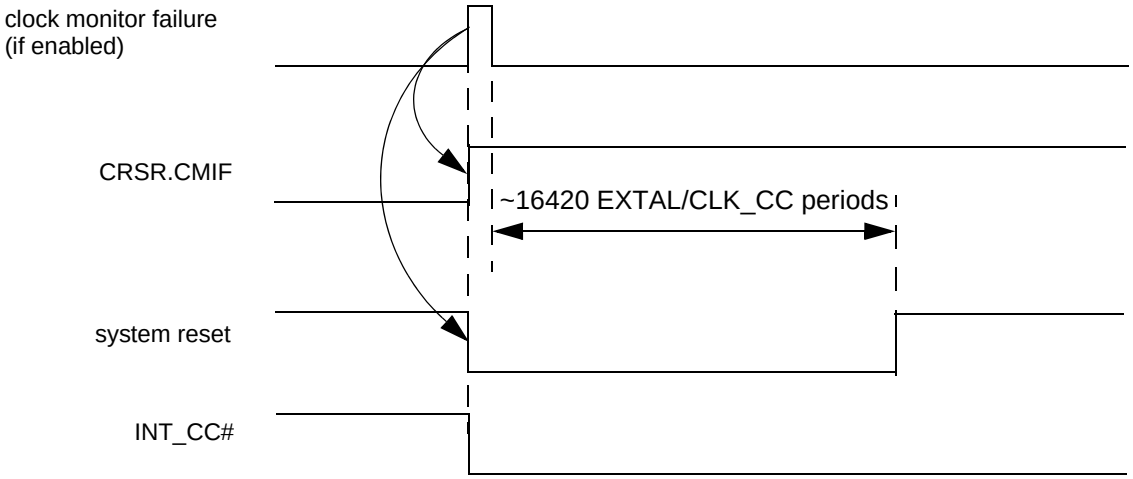


Figure 6-5. Clock Monitor Failure Reset

6.4.1.3 External Reset

When the *RESET#* signal is asserted the CRG asserts the system reset signal. The CRG deasserts the system reset signal approximately 70 *EXTAL/CLK_CC* clock periods after the deassertion of the *RESET#*.

The CRG asserts the *INT_CC#* interrupt line and the external reset interrupt flag, *CRSR.ERIF,* on the assertion of the *RESET#* signal.

NOTE

The CRG deasserts the INT_CC# signal when CRSR.PRIF, CRSR.LVIF, CRSR.CMIF and CRSR.ERIF bits are 0.

Figure 6-6 illustrates an external reset scheme.

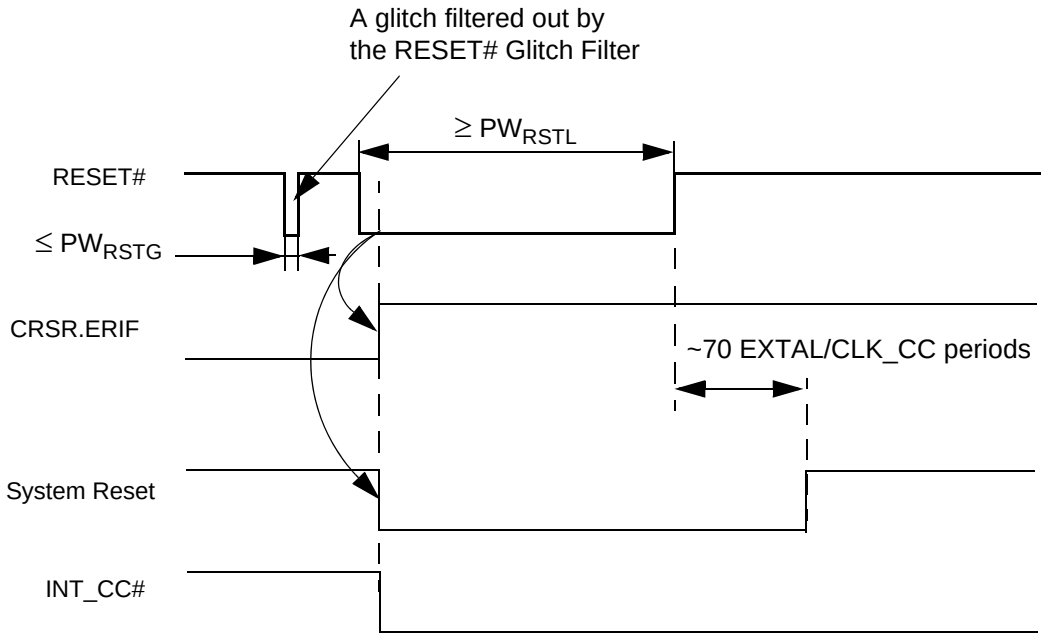


Figure 6-6. External Reset

6.4.1.4 RESET# Glitch Filter

The CRG has a built-in RESET# glitch filter to avoid device reset caused by glitches on the RESET# line. It removes glitches with durations less than or equal to PW_{RSTG}. Figure 6-6 illustrates an external reset sequence with a glitch filtered out by the glitch filter.

Timing characteristics of the glitch filter can be found in Table A-11.

6.4.2 Interface Selection

The interface mode selection is done when the *TXD_BG[1:2]/IF_SEL[1:0]* pins are in the *IF_SEL[1:0]* mode. In the *TXD_BG[1:2]* modes the pads are outputs from the MFR4310 device.

NOTE

The PIM block selects the TXD_BG[1:2]/IF_SEL[1:0] pads modes based on the system reset signal.

6.4.2.1 Interface and AMI Clock Selection

The interface selection is made upon the levels on the bus signal *IF_SEL[1:0]* while a power-on, low voltage, clock monitor (if enabled) or external reset process is ongoing. The CRG latches the *IF_SEL[1:0]* during the latching window as presented in Figure 6-7 and Figure 6-8.

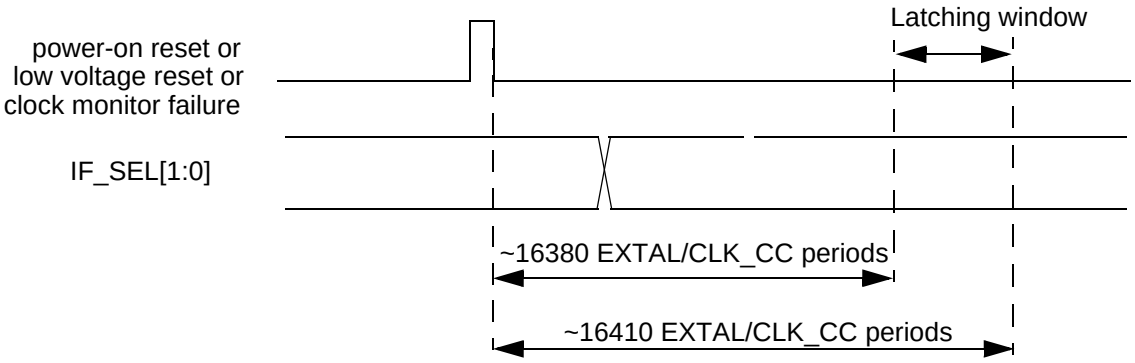


Figure 6-7. Interface Selection during Power-on or Low Voltage Reset or Clock Monitor Failure

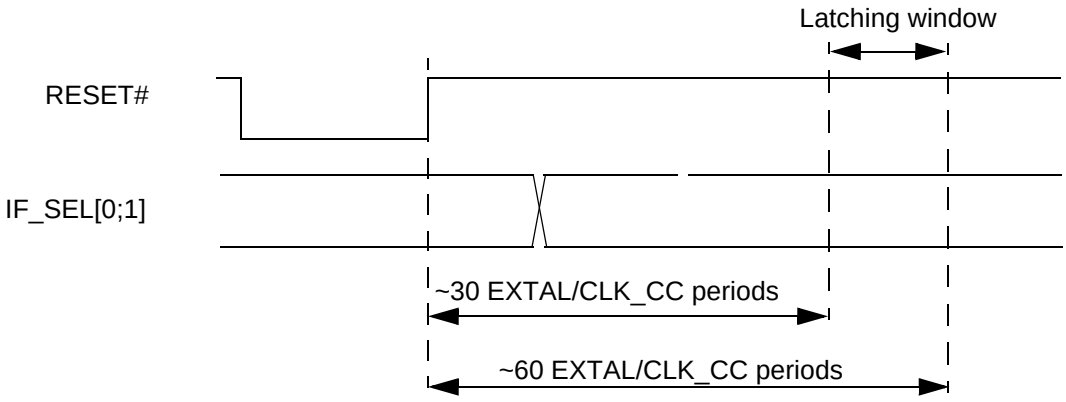


Figure 6-8. Interface Selection during External Reset

Table 6-5 shows the interface selection encoding provided by the CRSR.ECS bit:

Table 6-5. IF_SEL[1:0] Encoding by CRSR.ECS

IF_SEL1	IF_SEL0	CRSR.ECS
0	0	1
1	0	0
0	1	0
1	1	1

If, after the evaluation, the *IF_SEL[1:0]* are both low or both high, the CRG sets the CRSR.ECS bit to 1; otherwise the CRG resets the CRSR.ECS bit to 0.

6.4.3 CLKOUT Mode Selection and Control

The *CLKOUT* mode selection is done when the *DBG[3:2]/CLK_S[1:0]* pins are in the *CLK_S[1:0]* mode. In the *DBG[3:2]* modes the pads are outputs from the MFR4310 device.

NOTE

The PIM block selects the DBG[3:2]/CLK_S[1:0] pads modes based on the system reset signal.

The *CLKOUT* mode selection is made upon the levels of the *CLK_S[1:0]* signals in the latching window while a power-on, low voltage, clock monitor or external reset process is ongoing. The CRG latches the *CLK_S[1:0]* signal values during the latching window as presented on Figure 6-9, Figure 6-10 and Figure 6-11. The latched values are indicated in the *CRSR.CDCV* field.

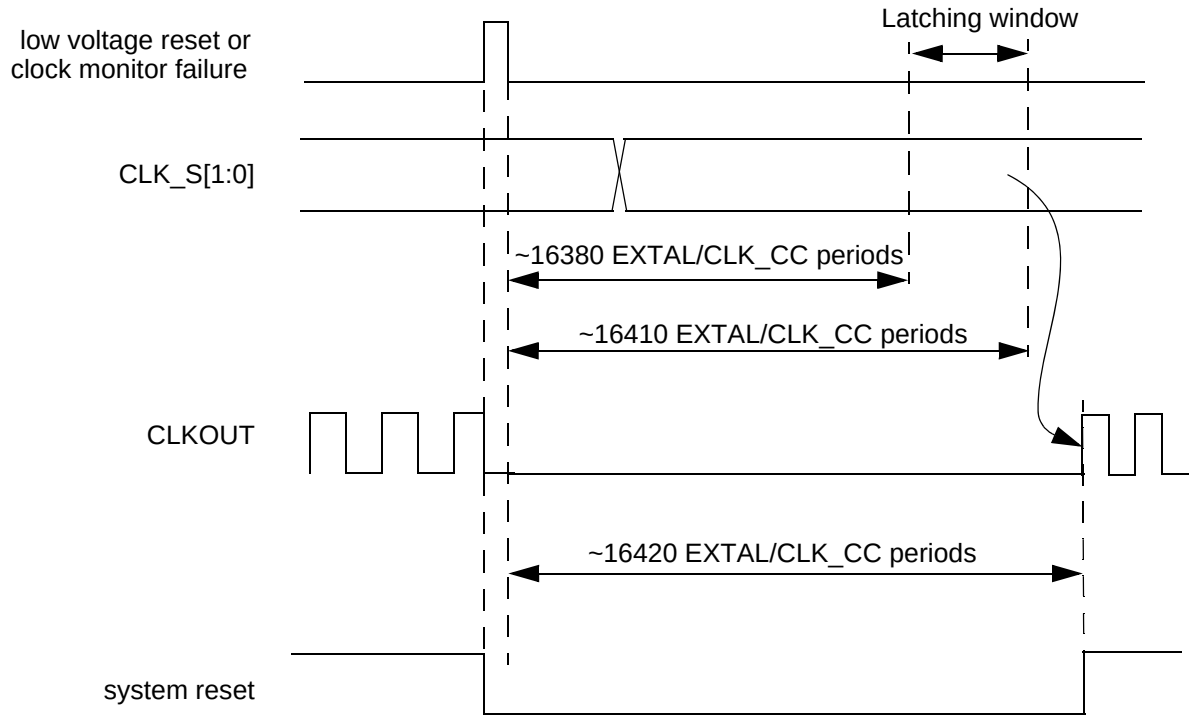


Figure 6-9. CLKOUT Mode Selection and Control during Low-voltage Reset or Clock Monitor Failure

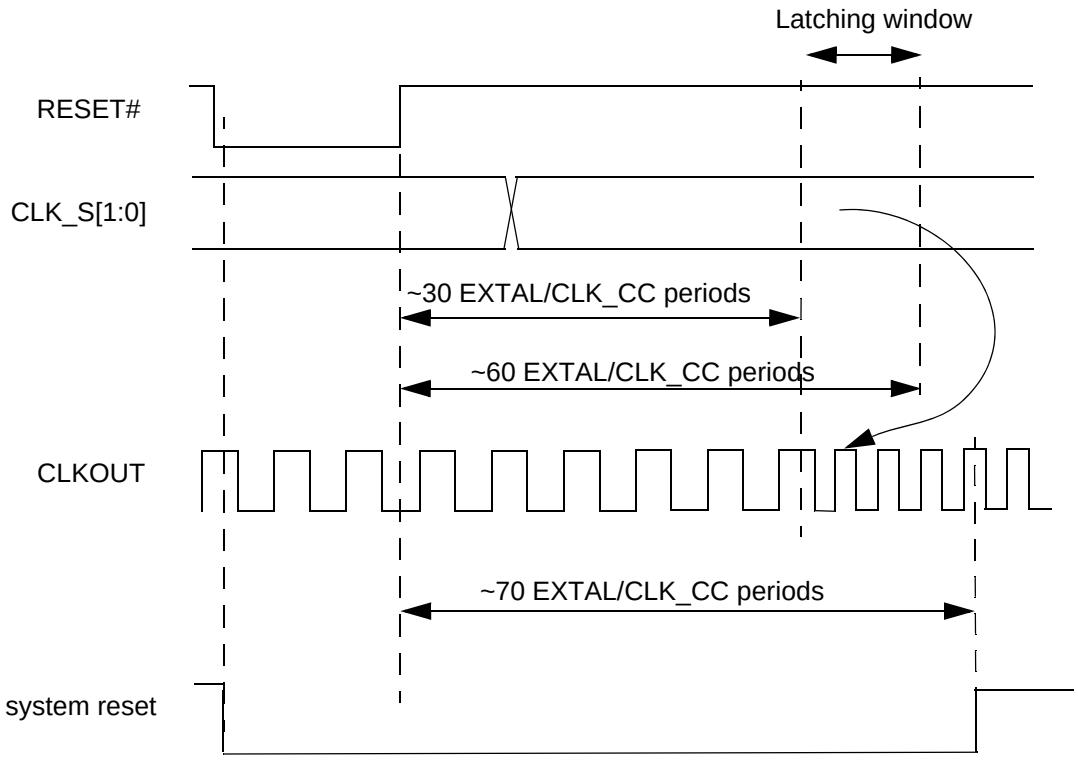


Figure 6-10. CLKOUT Mode Selection and Control during External Reset

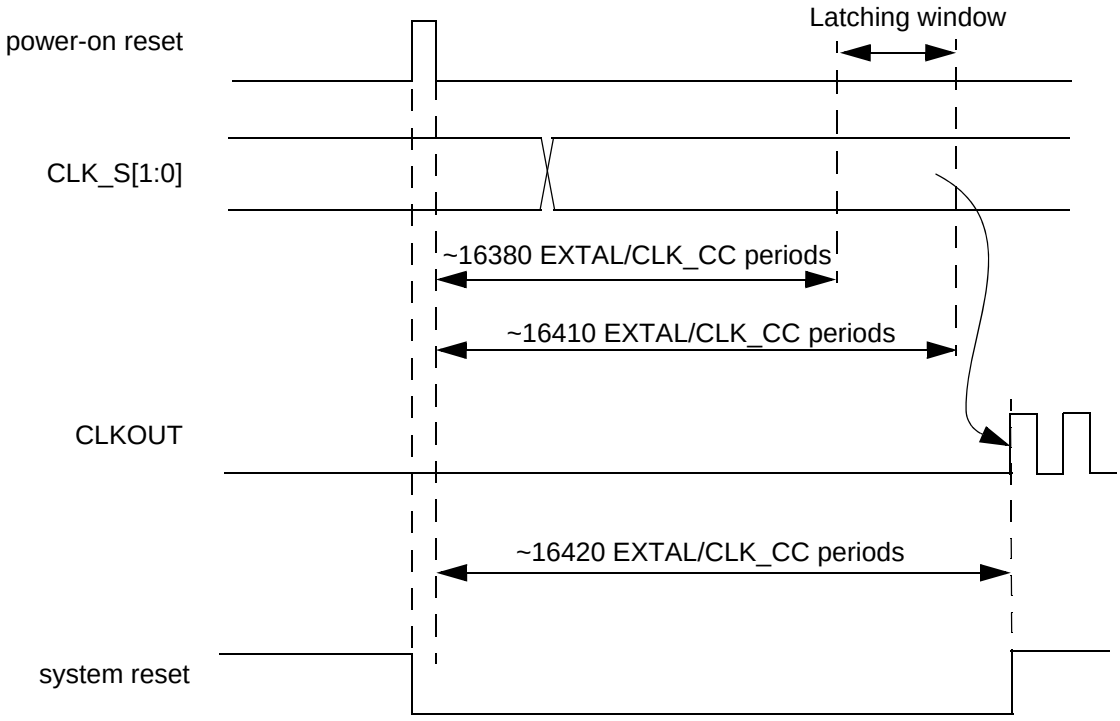


Figure 6-11. CLKOUT Mode Selection and Control during Power-on Reset

Chapter 7
Oscillator (OSCV2)

7.1 Introduction

The OSCV2 module provides one oscillator concept:

- A robust full swing Pierce oscillator with the possibility to feed in an external square wave

7.1.1 Features

The Pierce oscillator provides the following features:

- Wide high frequency operation range
- No DC voltage applied across the crystal
- Full rail-to-rail (2.5 V nominal) swing oscillation with low EM susceptibility
- Fast startup

Common features:

- Clock monitor (CM)
- Operation from the V_{DDOSC} 2.5 V (nominal) supply rail

7.1.2 Modes of Operation

One mode of operation exists:

- Full swing Pierce oscillator mode that can also be used to feed in an externally generated square wave suitable for high frequency operation and harsh environments

7.2 External Signal Description

This section lists and describes the signals that connect off chip.

7.2.1 V_{DDOSC} and V_{SSOSC} — OSC Operating Voltage, OSC Ground

These pins provide the operating voltage (V_{DDOSC}) and ground (V_{SSOSC}) for the OSCV2 circuitry. This allows the supply voltage to the OSCV2 to be independently bypassed.

7.2.2 EXTAL and XTAL — Clock/Crystal Source Pins

These pins provide the interface for a crystal or a CMOS compatible clock to control the internal clock generator circuitry. EXTAL is the external clock input or the input to the crystal oscillator amplifier. XTAL

is the output of the crystal oscillator amplifier. All internal system clocks are derived from the EXTAL input frequency.

NOTE

Freescale Semiconductor recommends an evaluation of the application board and chosen resonator or crystal by the resonator or crystal supplier.

The Crystal circuit is changed from standard.

The Pierce circuit is not suited for overtone resonators and crystals without a careful component selection.

For more information, see the EXTAL pin description in Chapter 2.

7.3 Memory Map and Register Definition

The CRG contains the registers and associated bits for controlling and monitoring the OSCV2 module.

7.4 Functional Description

The OSCV2 block has two external pins, EXTAL and XTAL. The oscillator input pin, EXTAL, is intended to be connected to a crystal or an external clock source. The XTAL pin is an output signal that provides crystal circuit feedback.

A buffered EXTAL signal, OSCCLK, becomes the internal reference clock. To improve noise immunity, the oscillator is powered by the V_{DDOSC} and V_{SSOSC} power supply pins.

7.4.1 Clock Monitor (CM)

The clock monitor circuit is based on an internal resistor-capacitor (RC) time delay so that it can operate without a clock. If no OSCCLK edges are detected within this RC time delay, the clock monitor indicates a failure which asserts self clock mode or generates a system reset depending on the state of the SCME bit. If the clock monitor is disabled or the presence of clocks is detected, no failure is indicated. The clock monitor function is enabled/disabled by the CME control bit, described in Chapter 6, “Clocks and Reset Generator (CRG)”.

7.5 Resets

OSCV2 contains a clock monitor, which can trigger a reset. The control bits and status bits for the clock monitor are described in Chapter 6, “Clocks and Reset Generator (CRG)”.

Appendix A
Electrical Characteristics

A.1 General

NOTE

The electrical characteristics given in this appendix are preliminary and must be used as a guide only. Values cannot be guaranteed by Freescale and are subject to change without notice.

NOTE

The part is specified and tested over the 5 V and 3.3 V ranges. For the intermediate range, generally the electrical specifications for the 3.3 V range apply, but the part is not tested in production test in the intermediate range.

This appendix provides the most accurate electrical information for the MFR4310 device available at the time of publication.

This introduction is intended to give an overview on several common topics like power supply, current injection etc.

A.1.1 Parameter Classification

The electrical parameters shown in this supplement are guaranteed by various methods. The following classifications are used and the parameters are tagged accordingly in the column labeled C in the parameter tables, where appropriate.

P: Parameters that are guaranteed during production testing on each individual device.

C: Parameters that are achieved by the design characterization by measuring a statistically relevant sample size across process variations.

T: Parameters that are achieved by design characterization on a small sample size from typical devices under typical conditions unless otherwise noted. All values shown in the typical column are within this category.

D: Parameters that are derived mainly from simulations.

A.1.2 Power Supply

The MFR4310 uses several pins to supply power to the I/O pins, oscillator and the digital core.

The VDDA, VSSA pair supplies the internal voltage regulator.

The VDDX, VSSX, VDDR and VSSR pairs supply the I/O pins, VDDR supplies also the internal voltage regulator.

VDD2_5 and VSS2_5 are the supply pins for the digital logic, VDDOSC, VSSOSC supply the oscillator.

VDDA, VDDX, VDDR as well as VSSA, VSSX, VSSR are connected by anti-parallel diodes for ESD protection.

NOTE

In the following context, VDD5 is used for VDDA, VDDR, and VDDX; VSS5 is used for VSSA, VSSR, and VSSX unless otherwise noted.

IDD5 denotes the sum of the currents flowing into the VDDA, VDDX, and VDDR pins.

VDD is used for VDD2_5 and VDDOSC; VSS is used for VSS2_5 and VSSOSC.

IDD is used for the current flowing into VDD2_5.

A.1.3 Pins

There are four groups of functional pins.

A.1.3.1 3.3V I/O pins

Those I/O pins have a nominal level of 3.3V. This class of pins is comprised of all I/O pins (all MFR4310 pins excluding EXTAL, XTAL and all power supply pins).The internal structure of all those pins is identical, however some of the functionality may be disabled. E.g. for the input-only pins the output drivers are disabled permanently.

A.1.3.2 Oscillator

The pins EXTAL, XTAL dedicated to the oscillator have a nominal 2.5V level. They are supplied by VDDOSC.

A.1.3.3 VDDR

This pin is used to enable the on chip voltage regulator.

A.1.4 Current Injection

Power supply must maintain regulation within operating V_{DD5} or V_{DD} range during instantaneous and operating maximum current conditions. If positive injection current ($V_{in} > V_{DD5}$) is greater than I_{DD5}, the

injection current may flow out of VDD5 and could result in external power supply going out of regulation. Ensure external V_{DD5} load shunts current greater than maximum injection current. This is the greatest risk when the CC is not consuming power; e.g. if no system clock is present, or if clock rate is very low which would reduce overall power consumption.

A.1.5 Absolute Maximum Ratings

CAUTION

Long-term exposure to absolute maximum ratings may affect device reliability, and permanent damage may occur if operate exceeding the rating. The device should be operated under recommended operating condition.

Absolute maximum ratings are stress ratings only. A functional operation under or outside those maxima is not guaranteed. Stress beyond those limits may affect the reliability or cause permanent damage of the device.

This device contains circuitry protecting against damage due to high static voltage or electrical fields; however, it is advised that normal precautions be taken to avoid application of any voltages higher than maximum-rated voltages to this high-impedance circuit. Reliability of operation is enhanced if unused inputs are tied to an appropriate logic voltage level (V_{SS5} or V_{DD5}).

Table A-1. Absolute Maximum Ratings

Num	Rating	Symbol	Min	Max	Unit
1	I/O, Regulator and Analog Supply Voltage	V_{DD5}	-0.3	6.5	V
2	Digital Logic Supply Voltage [1]	V_{DD}	-0.3	3.0	V
3	Oscillator Supply Voltage [1]	V_{DDOSC}	-0.3	3.0	V
4	Voltage difference VDDX to VDDR and VDDA	Δ_{VDDX}	-0.3	0.3	V
5	Voltage difference VSSX to VSSR and VSSA	Δ_{VSSX}	-0.3	0.3	V
6	Digital I/O Input Voltage[2]	V_{IN}	-0.3	6.5	V
7	EXTAL, XTAL inputs	V_{ILV}	-0.3	3.0	V
8	Instantaneous Maximum Current Single pin limit for all digital I/O pins [3]	I_D	-25	+25	mA
9	Instantaneous Maximum Current Single pin limit for EXTAL, XTAL[4]	I_{DL}	-25	+25	mA
10	Operating Temperature Range (packaged)	T_A	-40	+125	^oC
11	Operating Temperature Range (junction)	T_J	-40	+140	^oC
12	Storage Temperature Range	T_{stg}	– 65	+155	°C

1 The device contains an internal voltage regulator to generate the logic and OSC supply out of the I/O supply. The absolute maximum ratings apply when the device is powered from an external source.

2 AC over or undershoots for ±2V beyond the supply if limited to 20ns length are allowed.

3 All digital I/O pins are internally clamped to V_{SSX} and V_{DDX}, V_{SSR} and V_{DDR} or V_{SSA} and V_{DDA}.

4 Those pins are internally clamped to V_{SSOSC} and V_{DDOSC}.

A.1.6 ESD Protection and Latch-up Immunity

All ESD testing is in conformity with CDF-AEC-Q100 Stress test qualification for Automotive Grade Integrated Circuits. During the device qualification ESD stresses were performed for the Human Body Model (HBM), the Machine Model (MM) and the Charge Device Model.

A device is a failure if after exposure to ESD pulses the device no longer meets the device specification. Complete DC parametric and functional testing is performed per the applicable device specification at room temperature followed by hot temperature, unless specified otherwise in the device specification.

Table A-2. ESD and Latch-up Test Conditions

Model	Description	Symbol	Value	Unit
Human Body	Series Resistance	R1	1500	Ω
	Storage Capacitance	C	100	pF
	Number of Pulse per pin positive negative	-	- 3 3	
Machine	Series Resistance	R1	0	Ω
	Storage Capacitance	C	200	pF
	Number of Pulse per pin positive negative	-	- 3 3	
Latch-up	Minimum input voltage limit	-	-2.5	V
	Maximum input voltage limit	-	7.5	V

Table A-3. ESD and Latch-up Protection Characteristics

Num	C	Rating	Symbol	Min	Max	Unit
1	T	Human Body Model (HBM)	V_{HBM}	2000	-	V
2	T	Machine Model (MM)	V_{MM}	200	-	V
3	T	Charge Device Model (CDM)	V_{CDM}	500	-	V
4	T	Latch-up Current at T_A = 125°C positive negative	I_{LAT}	 +100 -100	-	mA
5	T	Latch-up Current at T_A = 27°C positive negative	I_{LAT}	 +200 -200	-	mA

A.1.7 Operating Conditions

This section describes the operating conditions of the device. Unless otherwise noted those conditions apply to all the following data.

NOTE

Refer to the temperature rating of the device (C, V, M) with regards to the ambient temperature T_A and the junction temperature T_J. For power dissipation calculations refer to Section A.1.8, "Power Dissipation and Thermal Characteristics".

Table A-4. Operating Conditions

Rating	Symbol	Min	Typ	Max	Unit
Oscillator and Quartz frequency[1]	f_{OSC}	-	40	40	MHz
Controller host interface clock frequency	f_{CHICLK_CC}	20	-	76	MHz
Quartz overtone		Fundamental Frequency			
Quartz frequency stability at T_J	f_{STB}	-1500	300	1500	ppm
Voltage difference VDDX to VDDR and VDDA	D_{VDDX}	-0.1	0	0.1	V
Voltage difference VSSX to VSSR and VSSA	D_{VSSX}	-0.1	0	0.1	V
I/O, Regulator and Analog Supply	V_{DD5}	2.97	3.3	5.5	V
Digital Logic Supply Voltage[2]	V_{DD}	2.25	2.5	2.75	V
Oscillator Supply Voltage[1]	V_{DDOSC}	2.25	2.5	2.75	V
Operating Junction Temperature Range	T_J	-40	-	+140	^oC
Operating Ambient Temperature Range[3]	T_A	-40	+27	+125	^oC

1 Input clock frequency applied to EXTAL/CLK_CC

2 The device contains an internal voltage regulator to generate the logic and OSC supply out of the I/O supply.

3 Refer to Section A.1.8, "Power Dissipation and Thermal Characteristics" for more information about the relation between ambient temperature T_A and device junction temperature T_J.

A.1.8 Power Dissipation and Thermal Characteristics

Power dissipation and thermal characteristics are closely related. The user must assure that the maximum operating junction temperature is not exceeded. The average chip-junction temperature (T_J) in °C can be obtained from:

$$T_J = T_A + (P_D \cdot \Theta_{JA}) \qquad \textit{Eqn. A-1}$$

T_J = Junction Temperature [°C]

T_A = Ambient Temperature [°C]

P_D = Total Chip Power Dissipation [W]

Θ_{JA} = Package Thermal Resistance [°C/W]

The total power dissipation can be calculated from:

$$P_D = P_{INT} + P_{IO} \quad \textit{Eqn. A-2}$$

P_{INT} = Chip Internal Power Dissipation [W]

Two cases with internal voltage regulator enabled and disabled must be considered:

1. Internal Voltage Regulator disabled

$$P_{INT} = I_{DD} \cdot V_{DD} + I_{DDOSC} \cdot V_{DDOSC} + I_{DDA} \cdot V_{DDA} \quad \textit{Eqn. A-3}$$

$$P_{IO} = \sum_i R_{DSON} \cdot I_{IO_i}^2 \quad \textit{Eqn. A-4}$$

P_{IO} is the sum of all output currents on I/O ports associated with VDDX and VDDR.
For R_{DSON} is valid:

$$R_{DSON} = \frac{V_{OL}}{I_{OL}}; \quad \text{for outputs driven low} \quad \textit{Eqn. A-5}$$

respectively

$$R_{DSON} = \frac{V_{DD5} - V_{OH}}{I_{OH}}; \quad \text{for outputs driven high} \quad \textit{Eqn. A-6}$$

2. Internal voltage regulator enabled

$$P_{INT} = I_{DDR} \cdot V_{DDR} + I_{DDA} \cdot V_{DDA} \quad \textit{Eqn. A-7}$$

I_{DDR} is the current shown in Table A-8 and not the overall current flowing into VDDR, which additionally contains the current flowing into the external loads with output high.

$$P_{IO} = \sum_i R_{DSON} \cdot I_{IO_i}^2 \quad \textit{Eqn. A-8}$$

P_{IO} is the sum of all output currents on I/O ports associated with VDDX and VDDR.

Table A-5. Thermal Package Simulation Details

Num	Rating	Symbol	Value	Unit
1	Junction to Ambient LQFP64, single sided PCB[1], Natural Convection	$R_{\theta JA}$	67	$^{o}C/W$
2	Junction to Ambient LQFP64, double sided PCB with 2 internal planes[1], Natural Convection	$R_{\theta JMA}$	48	$^{o}C/W$
3	Junction to Ambient LQFP64 (@200 ft/min), single sided PCB[1]	$R_{\theta JMA}$	55	$^{o}C/W$
4	Junction to Ambient LQFP64 (@200 ft/min), double sided PCB with 2 internal planes[1]	$R_{\theta JMA}$	42	$^{o}C/W$
5	Junction to Board LQFP64[2]	$R_{\theta JB}$	31	$^{o}C/W$
6	Junction to Case LQFP64[3]	$R_{\theta JC}$	14	$^{o}C/W$
7	Junction to Package Top LQFP64[4], Natural Convection	Ψ_{JT}	3	$^{o}C/W$

1 Junction-to-Ambient Thermal Resistance determined per JEDEC JESD51-3 and JESD51-6. Thermal test board meets JEDEC specification for this package.

2 Junction-to-Board thermal resistance determined per JEDEC JESD51-8. Thermal test board meets JEDEC specification for the specified package.

3 Junction-to-Case at the top of the package determined using MIL-STD 883 Method 1012.1. The cold plate temperature is used for the case temperature. Reported value includes the thermal resistance of the interface layer.

4 Thermal characterization parameter indicating the temperature difference between the package top and the junction temperature per JEDEC JESD51-2. When Greek letters are not available, the thermal characterization parameter is written as Psi-JT

A.1.9 I/O Characteristics

This section describes the characteristics of all 3.3V I/O pins. All parameters are not always applicable, e.g. not all pins feature pullup/pulldown resistances.

Table A-6. 5V I/O Characteristics (V_{DD5} = 5V)

Conditions are shown in Figure A-4, unless otherwise noted.

Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	P	Input High Voltage	V_{IH}	0.65*V_{DD5}	-	-	V
	T	Input High Voltage	V_{IH}	-	-	V_{DD5}+0.3	V
2	P	Input Low Voltage	V_{IL}	-	-	0.35*V_{DD5}	V
	T	Input Low Voltage	V_{IL}	V_{SS5}-0.3	-	-	V
3	C	Input Hysteresis	V_{HYS}	-	250	-	mV
4	P	High Impedance (Off-state) Leakage Current V_{IN}=V_{DD} or V_{SS}, all input/output and output pins	I_{IN}	-2.5	-	+2.5	uA
5	P	Output High Voltage (pins in output mode) @50% Partial Drive I_{OH} = -2mA	V_{OH}	V_{DD5}-0.8	-	-	V
6	P	Output High Voltage (pins in output mode) @100% Full Drive I_{OH} = -10mA	V_{OH}	V_{DD5}-0.8	-	-	V
7	P	Output Low Voltage (pins in output mode) @50% Partial Drive I_{OL} = +2mA	V_{OL}	-	-	0.8	V
8	P	Output Low Voltage (pins in output mode) @100% Full Drive I_{OL} = +10mA	V_{OL}	-	-	0.8	V
9	P	Internal Pullup Device Current, tested at V_{IL} Max	I_{PUL}	-	-	-130	uA
10	P	Internal Pullup Device Current, tested at V_{IH} Min.	I_{PUH}	-10	-	-	uA
11	P	Internal Pulldown Device Current, tested at V_{IH} Min.	I_{PDH}	-	-	130	uA
12	P	Internal Pulldown Device Current, tested at V_{IL} Max	I_{PDL}	10	-	-	uA
13	d	Input Capacitance (input, input/output pins)	C_{IN}	-	7	-	pF
14	T	Injection Current[1] Single Pin Limit Total Device Limit. Sum of all injected currents	 I_{ICS} I_{ICP}	 -2.5 -25	 - -	 2.5 25	mA
15	P	Load Capacitance 50% Partial Drive 100% Full Drive	C_L	-	-	 25 50	pF

[1] Refer to Section A.1.4, “Current Injection”, for more information.

Table A-7. 3.3V I/O Characteristics (V_{DD5} = 3.3V)

Conditions are V_{DDX}=3.3V +/-10% Temperature from -40°C to +140°C, unless otherwise noted

Num	C	Rating	Symbol	Min	Typ	Max	Unit
1	P	Input High Voltage	V_{IH}	0.65*V_{DD5}	-	-	V
	T	Input High Voltage	V_{IH}	-	-	V_{DD5}+0.3	V
2	P	Input Low Voltage	V_{IL}	-	-	0.35*V_{DD5}	V
	T	Input Low Voltage	V_{IL}	V_{SS5}-0.3	-	-	V
3	C	Input Hysteresis	V_{HYS}	-	250	-	mV
4	P	High Impedance (Off-state) Leakage Current V_{IN}=V_{DD} or V_{SS}, all input/output and output pins	I_{IN}	-2.5	-	+2.5	uA
5	P	Output High Voltage (pins in output mode) @50% Partial Drive I_{OH} = -0.75mA	V_{OH}	V_{DD5}-0.4	-	-	V
6	P	Output High Voltage (pins in output mode) @100% Full Drive I_{OH} = -4.5mA	V_{OH}	V_{DD5}-0.4	-	-	V
7	P	Output Low Voltage (pins in output mode) @50% Partial Drive I_{OL} = +0.9mA	V_{OL}	-	-	0.4	V
8	P	Output Low Voltage (pins in output mode) @100% Full Drive I_{OL} = +5.5mA	V_{OL}	-	-	0.4	V
9	P	Internal Pullup Device Current, tested at V_{IL} Max	I_{PUL}	-	-	-60	uA
10	P	Internal Pullup Device Current, tested at V_{IH} Min.	I_{PUH}	-6	-	-	uA
11	P	Internal Pulldown Device Current, tested at V_{IH} Min.	I_{PDH}	-	-	60	uA
12	P	Internal Pulldown Device Current, tested at V_{IL} Max	I_{PDL}	6	-	-	uA
13	D	Input Capacitance (input, input/output pins)	C_{IN}	-	7	-	pF
14	T	Injection Current[1] Single Pin Limit Total Device Limit. Sum of all injected currents	 I_{ICS} I_{ICP}	 -2.5 -25	 - -	 2.5 25	mA
15	P	Load Capacitance 50% Partial Drive 100% Full Drive	C_L	-	-	 25 50	pF

[1] Refer to Section A.1.4, “Current Injection” for more information.

A.1.10 Supply Currents

All measurements are done without output loads. Unless otherwise noted, the currents are measured with internal voltage regulator enabled and a 40 MHz oscillator, in standard Pierce mode. Production testing is performed using a square wave signal at the EXTAL input.

Table A-8. Supply Current Characteristics

Conditions are shown in Table A-4 unless otherwise noted								
Num	**C**	**Rating**		**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	P	Run supply currents Internal regulator enabled	-40°C	I_{DD5}	-	-	50	mA
			25°C		-	-	50	
			140°C		-	-	50	

A.2 Voltage Regulator (VREG)

A.2.1 Operating Conditions

Table A-9. Voltage Regulator — Operating Conditions

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Characteristic**	**Symbol**	**Min**	**Typical**	**Max**	**Unit**
1	P	Input Voltages	$V_{VDDR,A}$	2.97	—	5.5	V
2	P	Regulator Current Shutdown Mode	I_{REG}	—	—	40	μA
3	P	Output Voltage Core Full Performance Mode Shutdown Mode	V_{DD}	2.45 —	2.5 —[1]	2.75 —	V V
4	P	Output Voltage OSC Full Performance Mode Shutdown Mode	V_{DDOSC}	2.35 —	2.5 —[2]	2.75 —	V V
5	P	Low Voltage Reset[3] Assert Level	V_{LVRA}	2.25	—	—	V
6	C	Power-on Reset[4] Assert Level Deassert Level	V_{PORA} V_{PORD}	0.97 —	— —	— 2.07	V V

1 High Impedance Output

2 High Impedance Output

3 Monitors V_{DD}, always active

4 Monitors V_{DD}, always active

A.2.2 Chip Power-up and Voltage Drops

Voltage regulator sub modules POR (power-on reset) and LVR (low voltage reset) handle chip power-up or drops of the supply voltage. Their function is described in Figure A-1.

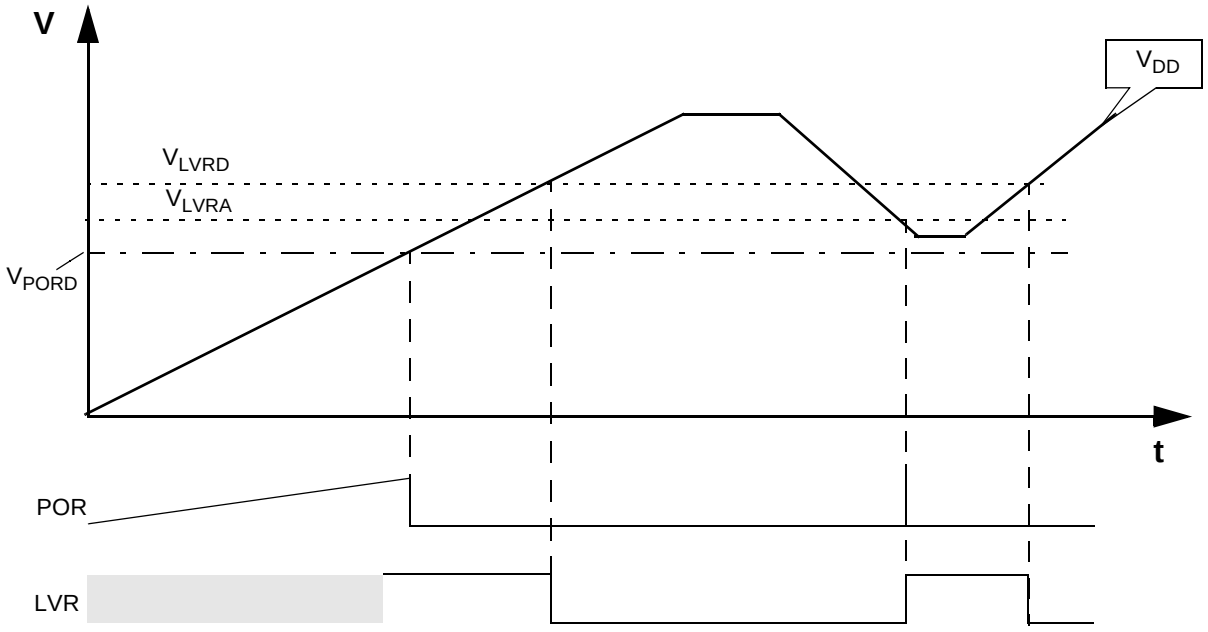


Figure A-1. Voltage Regulator — Chip Power-up and Voltage Drops (not scaled)

A.2.3 Output Loads

A.2.3.1 Resistive Loads

On-chip voltage regulator intended to supply the internal logic and oscillator circuits allows no external DC loads.

A.2.3.2 Capacitive Loads

The capacitive loads are specified in Figure A-10. Ceramic capacitors with X7R dielectricum are required.

Table A-10. Voltage Regulator Recommended Capacitive Loads

Num	Characteristic	Symbol	Min	Typical	Max	Unit
1	VDD external capacitive load	C_{DDext}	200	440	12000	nF
3	VDDOSC external capacitive load	$C_{DDOSCext}$	90	220	5000	nF

A.3 Reset and Oscillator

This section summarizes the electrical characteristics of the various startup scenarios for the Oscillator.

A.3.1 Startup

Table A-11 summarizes several startup characteristics explained in this section. Detailed description of the startup behavior can be found in Chapter 6, “Clocks and Reset Generator (CRG)”.

Table A-11. Startup Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	T	POR deassert level	V_{PORD}	-	-	2.07	V
2	T	POR assert level	V_{PORA}	0.97	-	-	V
3	D	Reset input pulse width, minimum input time	PW_{RSTL}	14	-	-	ns
4	D	Filtered glitch duration	PW_{RSTG}	-	-	3	ns

A.3.1.1 POR

The release level V_{PORD} (see Table A-9) and the assert level V_{PORA} (see Table A-9) are derived from the V_{DD} supply. They are also valid if the device is powered externally. After releasing the POR reset the oscillator is started.

A.3.1.2 LVR

The assert level V_{LVRA} (see Table A-9) is derived from the V_{DD} supply. After releasing the LVR reset, the oscillator is started.

A.3.1.3 External Reset

When external reset is asserted for a time greater than PW_{RSTL} the CRG module generates an internal reset, and the CC starts operations, if there was an oscillation before reset.

A.3.2 Oscillator

The device features an internal Pierce oscillator with a clock monitor. A clock monitor failure is asserted if the clock signal is below the Clock Monitor Assert Frequency, f_{CMAF}.

Table A-12. Oscillator Characteristics

Conditions are shown in Table A-4 unless otherwise noted							
Num	**C**	**Rating**	**Symbol**	**Min**	**Typ**	**Max**	**Unit**
1	C	Crystal oscillator range (Pierce)[1, 2]	f_{OSC}	0.5	-	40	MHz
2	P	Startup Current	i_{OSC}	100	-	-	μA
4	P	Clock monitor assert frequency	f_{CMAF}	50	100	200	kHz
5	P	External square wave input frequency	f_{EXT}	0.5	-	50	MHz
6	D	External square wave pulse width low	t_{EXTL}	9.5	-	-	ns
7	D	External square wave pulse width high	t_{EXTH}	9.5	-	-	ns
8	D	External square wave rise time	t_{EXTR}	-	-	1	ns
9	D	External square wave fall time	t_{EXTF}	-	-	1	ns
10	D	Input Capacitance (EXTAL, XTAL pins)	C_{IN}	-	7	-	pF
11	C	DC Operating Bias in Pierce mode on EXTAL Pin	V_{DCBIAS}	-	2.5	-	V

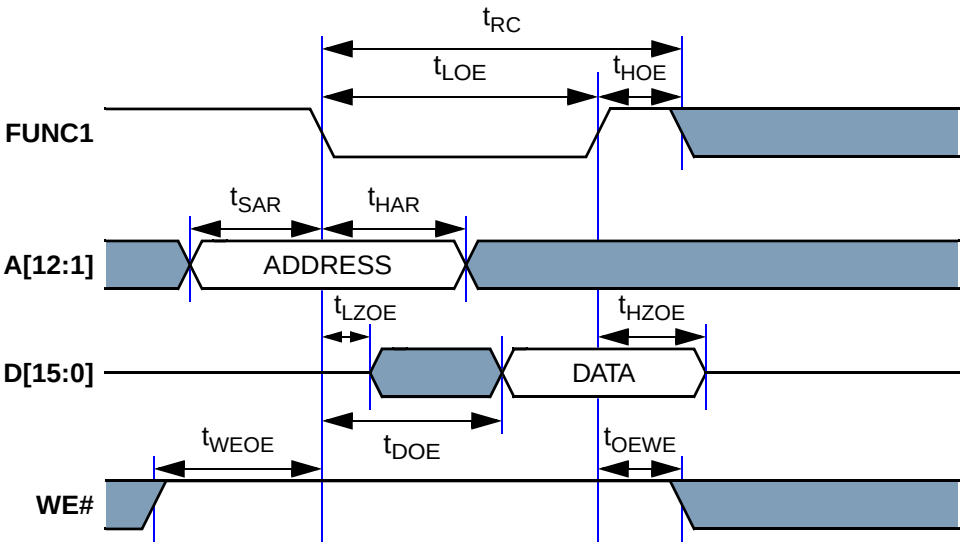
1 Depending on the crystal, a damping series resistor might be necessary.

2 Input clock frequency applied to EXTAL/CLK_CC.

A.4 Asynchronous Memory Interface Timing

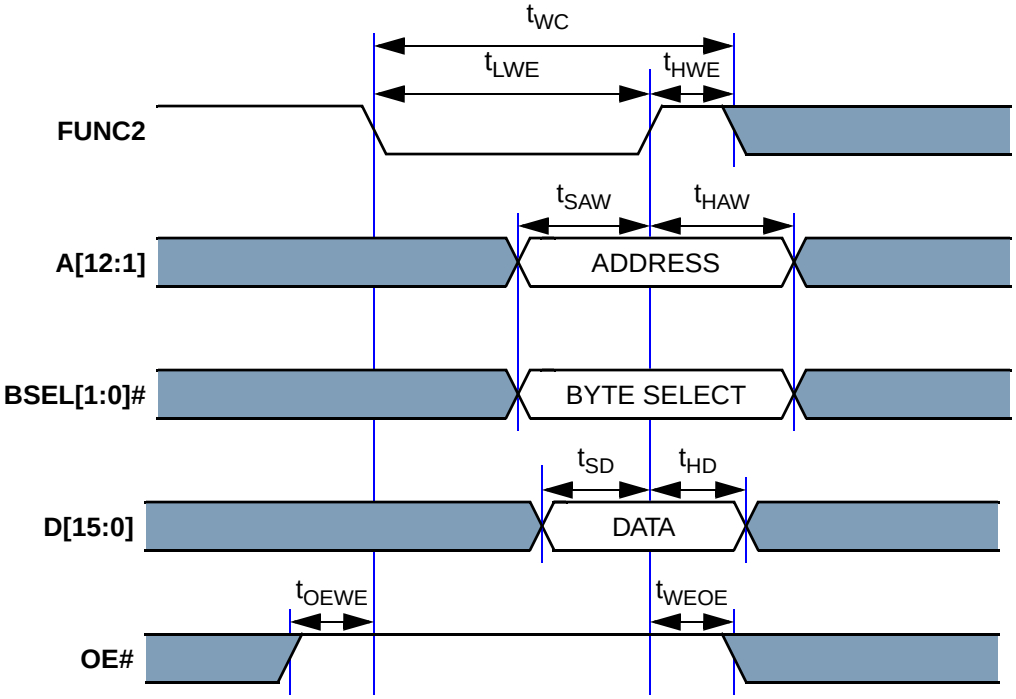
The CC AMI Interface read/write timing diagram is shown in the following figures.

- Writing to the device is accomplished when Chip Enable (CE#) and Write Enable (WE#) inputs are LOW (asserted).
- Reading from the device is accomplished when Chip Enable (CE#) and Output Enable (OE#) are LOW (asserted) while the Write Enable (WE#) is HIGH (deasserted).
- The input/output pins D[15:0] are in a high-impedance state when the device is not selected (CE# is HIGH), the outputs are disabled (OE# HIGH) or during a write operation (CE# LOW, and WE# LOW).



Note: The signal FUNC1 is a logical OR of the chip enable (CE#) and output enable (OE#) inputs.

Figure A-2. AMI Interface Read Timing Diagram



Note: The signal FUNC2 is a logical OR of the chip enable (CE#) and write enable (WE#) inputs.

Figure A-3. AMI Interface Write Timing Diagram

Table A-13. AMI Interface AC Switching Characteristics Over the Operating Range[1]

Characteristic	Symbol	Min	Max	Unit
Read Cycle				
Read Time Cycle	t_{RC}	$2.5 \times t_{AMI\_CLK} + 32$		ns
Address Setup Read	t_{SAR}	5		ns
Address Hold Read	t_{HAR}	5		ns
OE# LOW to Data valid	t_{DOE}		$2.5 \times t_{AMI\_CLK} + 23$	ns
OE# LOW time	t_{LOE}	$2.5 \times t_{AMI\_CLK} + 27$[2]		ns
OE# HIGH time	t_{HOE}	5		ns
OE# LOW to Low-Z	t_{LZOE}	5		ns
OE# HIGH to High-Z	t_{HZOE}		15	ns
WE# HIGH to OE# LOW	t_{WEOE}	$1 \times t_{AMI\_CLK}$		ns
Write Cycle				
Write Time Cycle	t_{WC}	$3 \times t_{AMI\_CLK} + 10$		ns
Address Setup Write	t_{SAW}	5		ns
Address Hold Write	t_{HAW}	5		ns
Data Setup	t_{SD}	5		ns
Data Hold	t_{HD}	5		ns
WE# LOW time	t_{LWE}	$1.5 \times t_{AMI\_CLK} + 5$		ns
WE# HIGH time	t_{HWE}	$0.5 \times t_{AMI\_CLK} + 5$		ns
OE# HIGH to WE# LOW	t_{OEWE}	0		ns

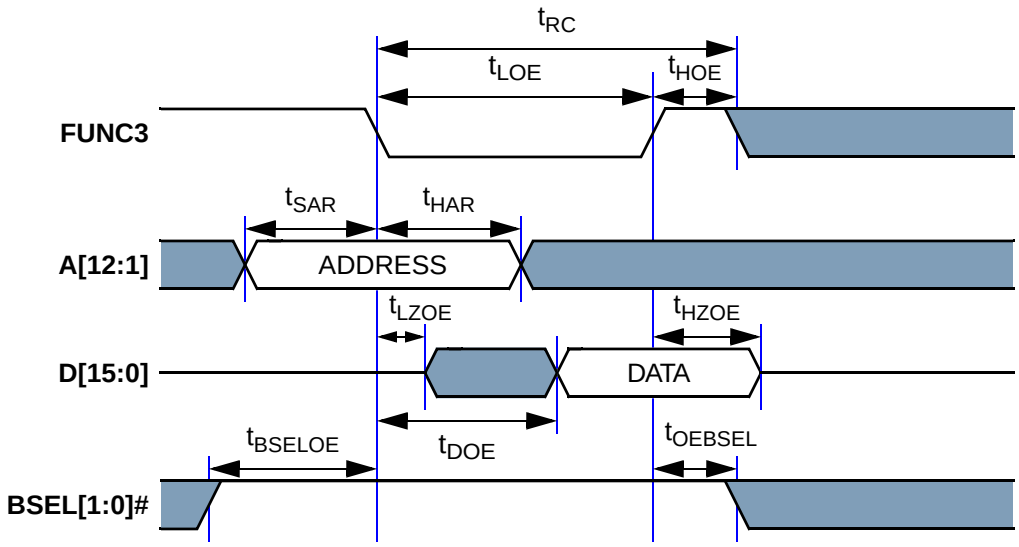
1 t_{AMI_CLK} is the period in ns of the CHI and host interface clock selected by IF_SEL[1:0] as described in Table 2-6.

2 Depends on duty cycle of the CHI and host interface clock: $t_{LOE} = (3.0 \times t_{AMI\_CLK}) - t_{AMI\_CLK\_HIGH} + 27$, where $t_{AMI\_CLK\_HIGH}$ is the period in ns of the high phase of the CHI and host interface clock.

A.5 MPC Interface Timing

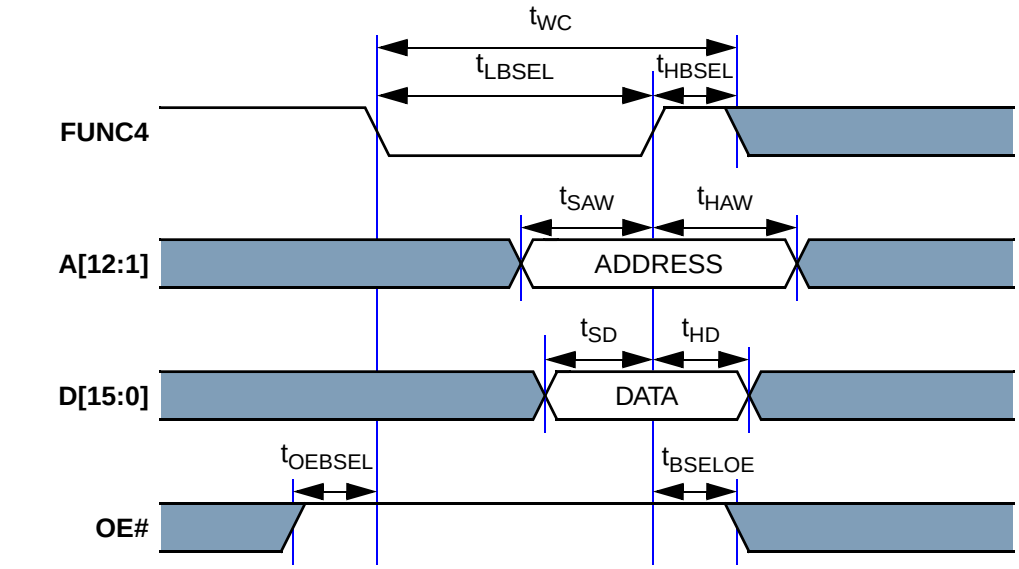
The CC MPC Interface read/write timing diagram is shown in the following figures.

- Writing to the device is accomplished when Chip Enable (CE#) and at least one of the Byte Selects (BSEL[1:0]#) inputs are LOW (asserted).
- Reading from the device is accomplished when Chip Enable (CE#) and Output Enable (OE#) is LOW (asserted) while both Byte Selects (BSEL[1:0]#) are HIGH (deasserted).
- The input/output pins (D[15:0]) are in a high-impedance state when the device is not selected (CE# is HIGH), the outputs are disabled (OE# HIGH) or during a write operation (CE# LOW and at least one BSEL[1:0]# LOW).



Note: The signal FUNC3 is a logical OR of the chip enable (CE#) and output enable (OE#) inputs.

Figure A-4. MPC Interface Read Timing Diagram



Note: The signal FUNC4 is a logical OR of the chip enable (CE#) and the logically ANDed byte select (BSEL[1:0]#) inputs.

Figure A-5. MPC Interface Write Timing Diagram

Table A-14. MPC Interface AC Switching Characteristics Over the Operating Range[1]

Characteristic	Symbol	Min	Max	Unit
Read Cycle				
Read Time Cycle	t_{RC}	$2.5 \times t_{CHICLK\_CC} + 32$		ns
Address Setup Read	t_{SAR}	5		ns
Address Hold Read	t_{HAR}	5		ns
OE# LOW to Data valid	t_{DOE}		$2.5 \times t_{CHICLK\_CC} + 23$	ns
OE# LOW time	t_{LOE}	$2.5 \times t_{CHICLK\_CC} + 27$[2]		ns
OE# HIGH time	t_{HOE}	5		ns
OE# LOW to Low-Z	t_{LZOE}	5		ns
OE# HIGH to High-Z	t_{HZOE}		15	ns
BSEL[1:0]# HIGH to OE# LOW	t_{BSELOE}	$1 \times t_{CHICLK\_CC}$		ns
Write Cycle				
Write Time Cycle	t_{WC}	$3 \times t_{CHICLK\_CC} + 10$		ns
Address Setup Write	t_{SAW}	5		ns
Address Hold Write	t_{HAW}	5		ns
Data Setup	t_{SD}	5		ns
Data Hold	t_{HD}	5		ns
BSEL[1:0]# LOW time	t_{LBSEL}	$1.5 \times t_{CHICLK\_CC} + 5$		ns
BSEL[1:0]# HIGH time	t_{HBSEL}	$0.5 \times t_{CHICLK\_CC} + 5$		ns
OE# HIGH to BSEL[1:0]# LOW	t_{OEBSEL}	0		ns

1 t_{CHICLK_CC} is the period in ns of the CHI and host interface clock CHICLK_CC.

2 Depends on duty cycle of the CHI and host interface clock: $t_{LOE} = (3.0 \times t_{CHICLK\_CC}) - t_{CHICLK\_CC\_HIGH} + 27$, where $t_{CHICLK\_CC\_HIGH}$ is the period in ns of the high phase of the CHI and host interface clock.

A.6 HCS12 Interface Timing

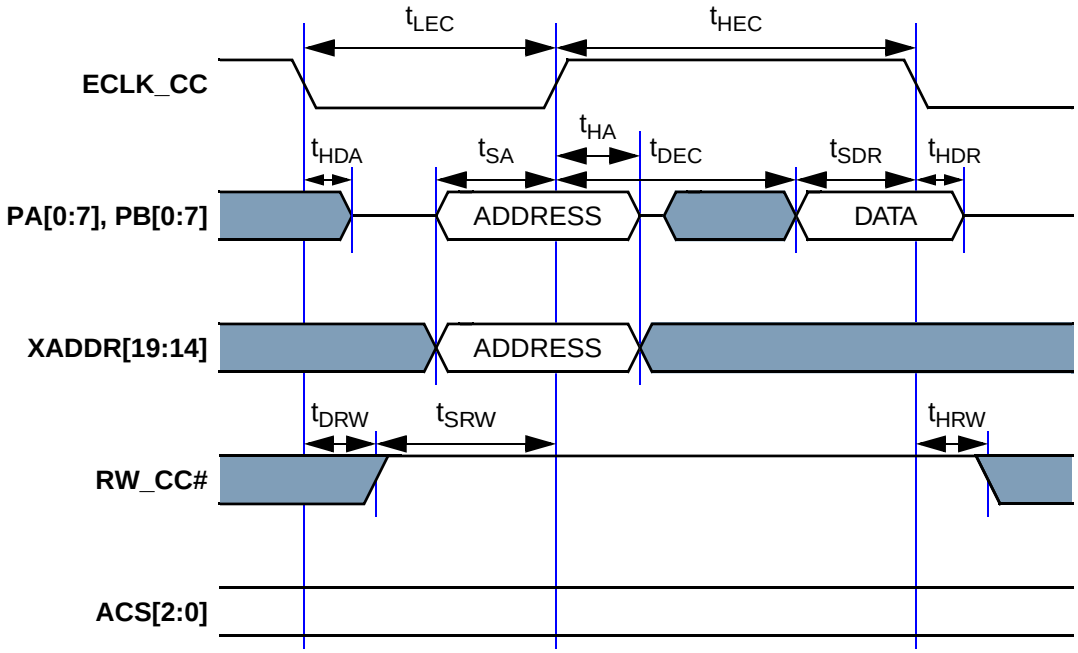


Figure A-6. HCS12 Interface Read Timing Diagram

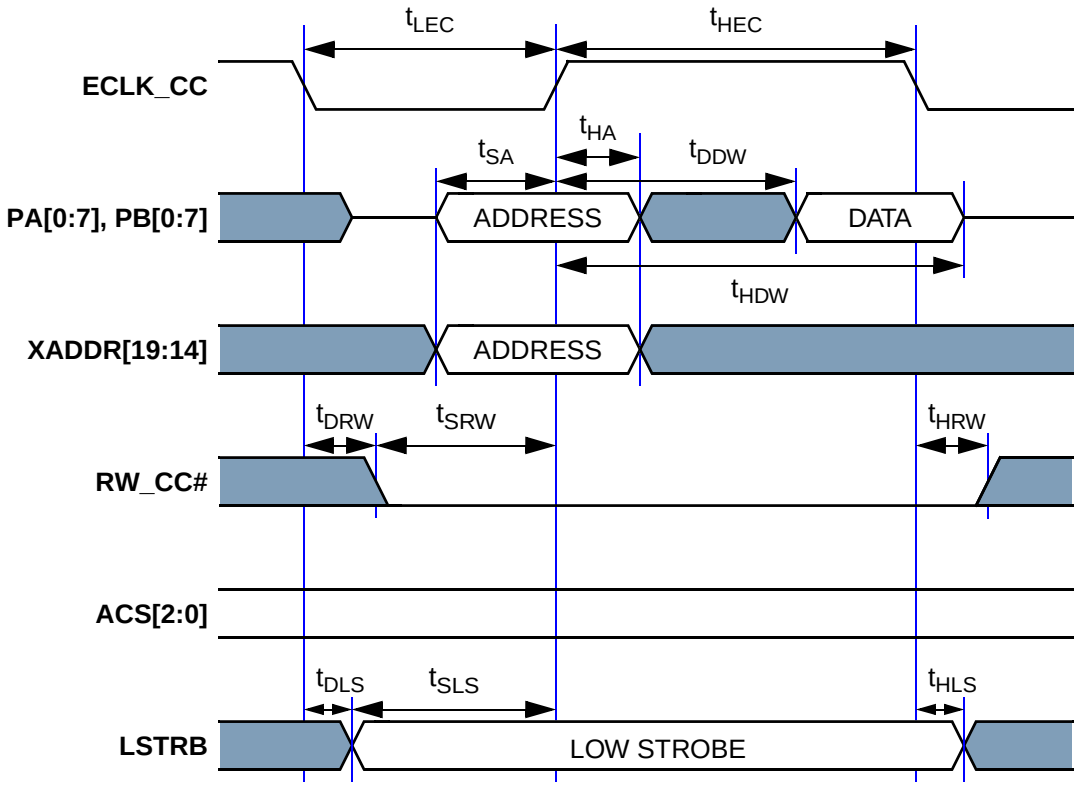


Figure A-7. HCS12 Interface Write Timing Diagram

Table A-15. HCS12 Interface AC Switching Characteristics Over the Operating Range[1]

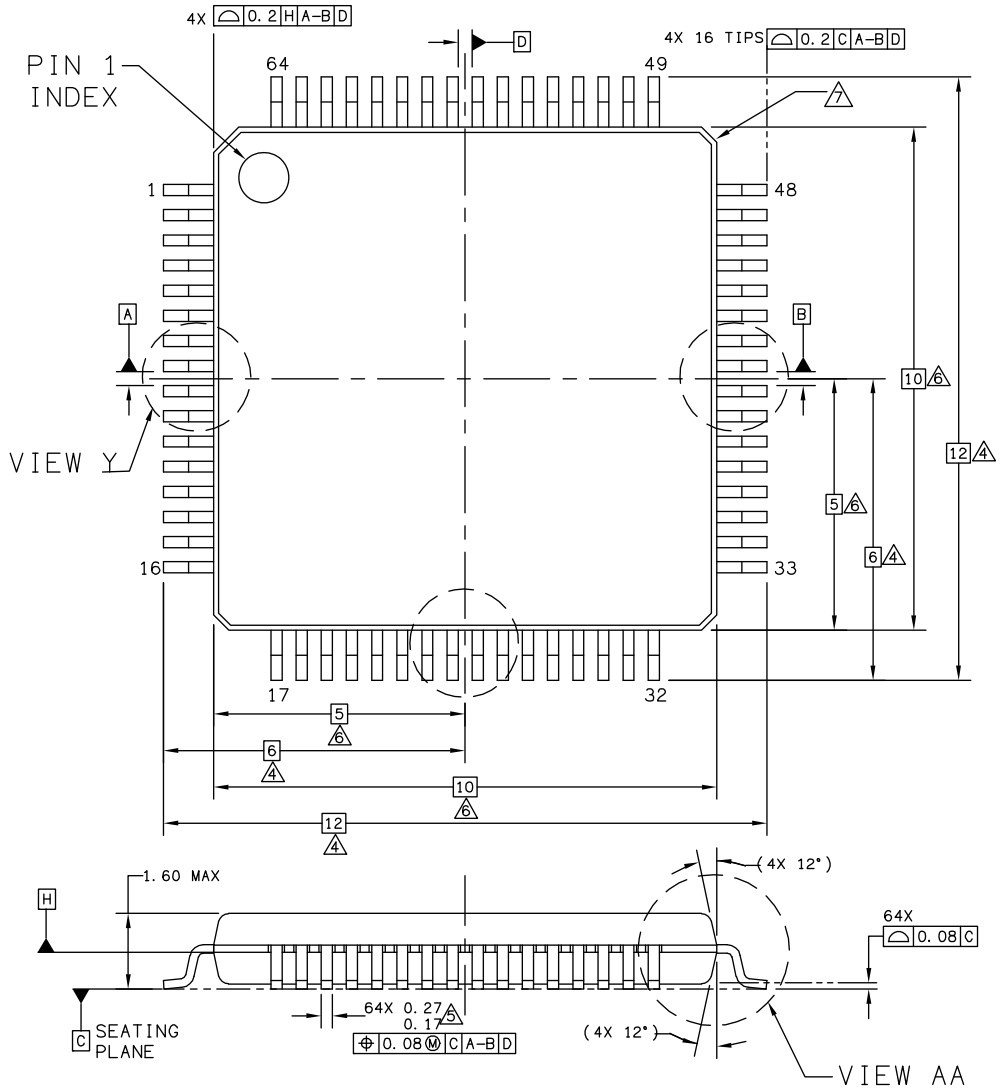
Characteristic	Symbol	Min	Max	Unit
Pulse width, ECLK_CC Low	t_{LEC}	30	-	ns
Pulse width, ECLK_CC High	t_{HEC}	99[2]	-	ns
Address valid time to ECLK_CC rise	t_{SA}	11	-	ns
Write Data delay time	t_{DDW}	-	70	ns
Write Data hold time	t_{HDW}	80		ns
RW_CC# delay time	t_{DRW}	-	7	ns
RW_CC# valid time to ECLK_CC rise	t_{SRW}	14	-	ns
RW_CC# hold time	t_{HRW}	2	-	ns
Data hold to address	t_{HDA}	2	-	ns
Multiplexed Address hold time	t_{HA}	2	-	ns
ECLK_CC high access time (ECLK_CC high to Read Data valid)	t_{DEC}	50	90	ns
Read Data setup time	t_{SDR}	13	-	ns
Read Data hold time	t_{HDR}	0	-	ns
Low strobe delay time	t_{DLS}	-	7	ns
Low strobe valid to ECLK_CC rise	t_{SLS}	14	-	ns
Low strobe hold time	t_{HLS}	2	-	ns

1 Based on f_{CLK_CC} = 40 MHz.

2 Depends on duty cycle of EXTAL/CLK_CC: $t_{HEC} = 99 + (t_{CLK\_CC} \times 0.5) - t_{CLK\_CC\_HIGH}$, where t_{CLK_CC} is the period in ns of EXTAL/CLK_CC and $t_{CLK\_CC\_HIGH}$ is the period in ns of the high phase of EXTAL/CLK_CC.

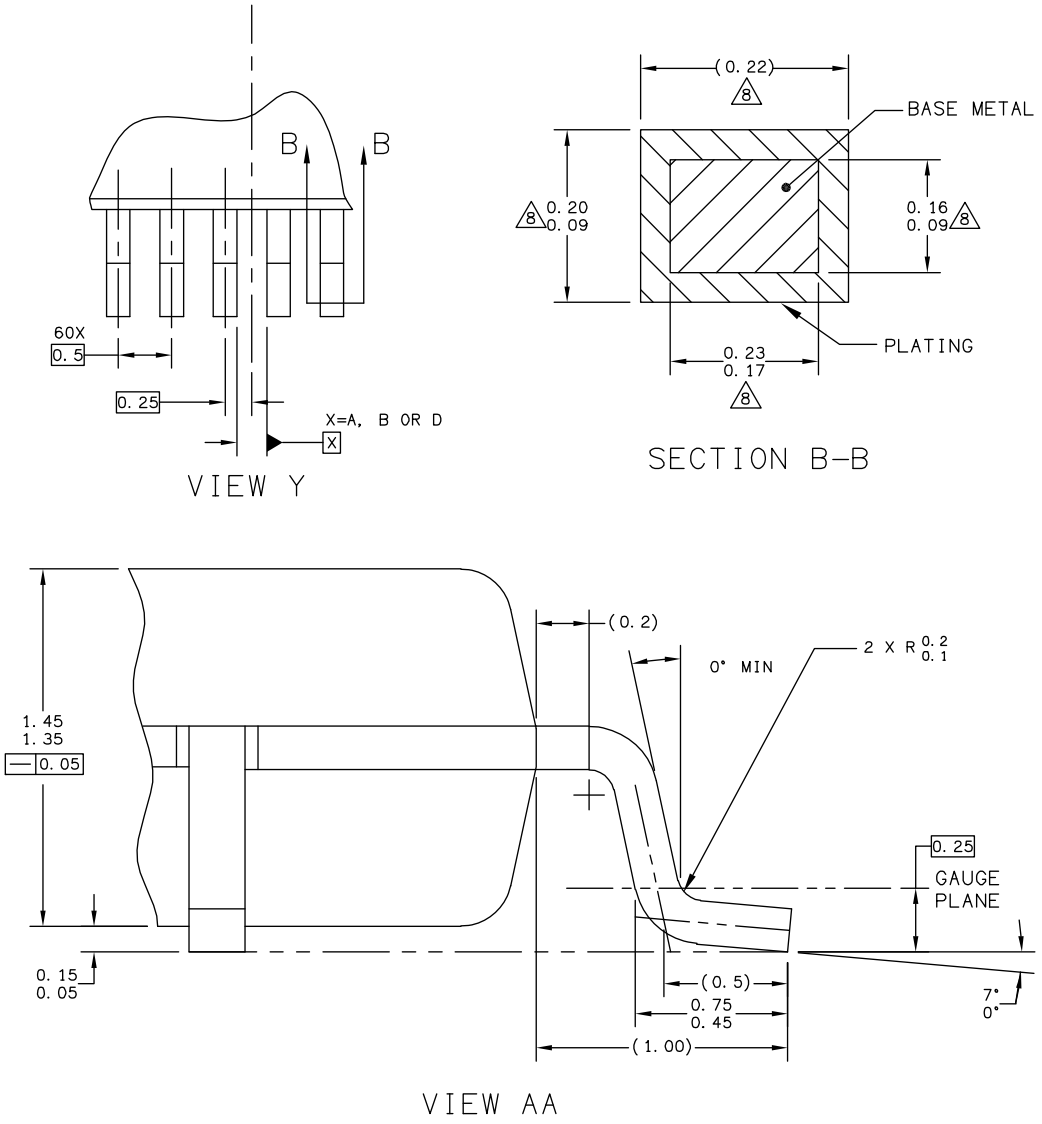
Appendix B
Package Information

B.1 64-pin LQFP package



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 64LD LQFP, 10 X 10 X 1.4 PKG, 0.5 PITCH, CASE OUTLINE	DOCUMENT NO: 98ASS23234W	REV: E
	CASE NUMBER: 840F-02	11 AUG 2006
	STANDARD: JEDEC MS-026 BCD	

Figure B-1. 64-pin LQFP Mechanical Dimensions (Case N 840F-02) (Page 1)



© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 64LD LQFP, 10 X 10 X 1.4 PKG, 0.5 PITCH, CASE OUTLINE	DOCUMENT NO: 98ASS23234W	REV: E
	CASE NUMBER: 840F-02	11 AUG 2006
	STANDARD: JEDEC MS-026 BCD	

Figure B-2. 64-pin LQFP Mechanical Dimensions (Case N 840F-02) (Page 2)

NOTES:

1. DIMENSIONS ARE IN MILLIMETERS.
2. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994.
3. DATUMS A, B AND D TO BE DETERMINED AT DATUM PLANE H.
4. DIMENSIONS TO BE DETERMINED AT SEATING PLANE C.
5. THIS DIMENSION DOES NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE UPPER LIMIT BY MORE THAN 0.08 mm AT MAXIMUM MATERIAL CONDITION. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD SHALL NOT BE LESS THAN 0.07 mm.
6. THIS DIMENSION DOES NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. THIS DIMENSION IS MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
7. EXACT SHAPE OF EACH CORNER IS OPTIONAL.
8. THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE LEAD BETWEEN 0.1 mm AND 0.25 mm FROM THE LEAD TIP.

© FREESCALE SEMICONDUCTOR, INC. ALL RIGHTS RESERVED.	MECHANICAL OUTLINE	PRINT VERSION NOT TO SCALE
TITLE: 64LD LQFP, 10 X 10 X 1.4 PKG, 0.5 PITCH, CASE OUTLINE	DOCUMENT NO: 98ASS23234W	REV: E
	CASE NUMBER: 840F–02	11 AUG 2006
	STANDARD: JEDEC MS–026 BCD	

Figure B-3. 64-pin LQFP Mechanical Dimensions (Case N 840F-02) (Page 3)

Appendix C
Printed Circuit Board Layout Recommendations

The PCB must be laid out carefully to ensure proper operation of the voltage regulator and the CC. The following rules must be observed:

- Every supply pair must be decoupled by a ceramic capacitor connected as near as possible to the corresponding pins (Cd).
- The central point of the groundstar should be the VSSR pin.
- Low-ohmic low-inductance connections should be used between VSSX and VSSR.
- VSSOSC must be directly connected to VSSR.
- Traces of VSSOSC, EXTAL and XTAL must be kept as short as possible. Occupied board area for C1, C2, C3 and Q should be as small as possible.
- Other signals or supply lines should not be routed under the area occupied by C1, C2, C3, and Q and the connection area of the CC.
- The central power input should be fed in at the VDDA/VSSA pins.

Figure C-1 shows a recommended PCB layout (64-pin LQFP) for standard Pierce oscillator mode, while Table C-1 provides suggested values for the external components.

Table C-1. Suggested External Component Values

Component	Purpose	Type	Value
C1	OSC load cap	ceramic X7R	2pF
C2	OSC load cap	ceramic X7R	2pF
C3	VDDOSC filter cap	ceramic X7R	100– 220nF
C4	VDDA filter cap	ceramic X7R	100– 220nF
Cd	VDDR, VDDX filter cap	ceramic X7R/tantalum	100– 220nF
Cload	VDD2_5 filter cap	ceramic X7R	100– 220nF
R_B	OSC resistance		1 MΩ
R_S	OSC resistance		0 Ω (i.e. short-circuit)
Q	Quartz	NDK NX8045GA	40 MHz

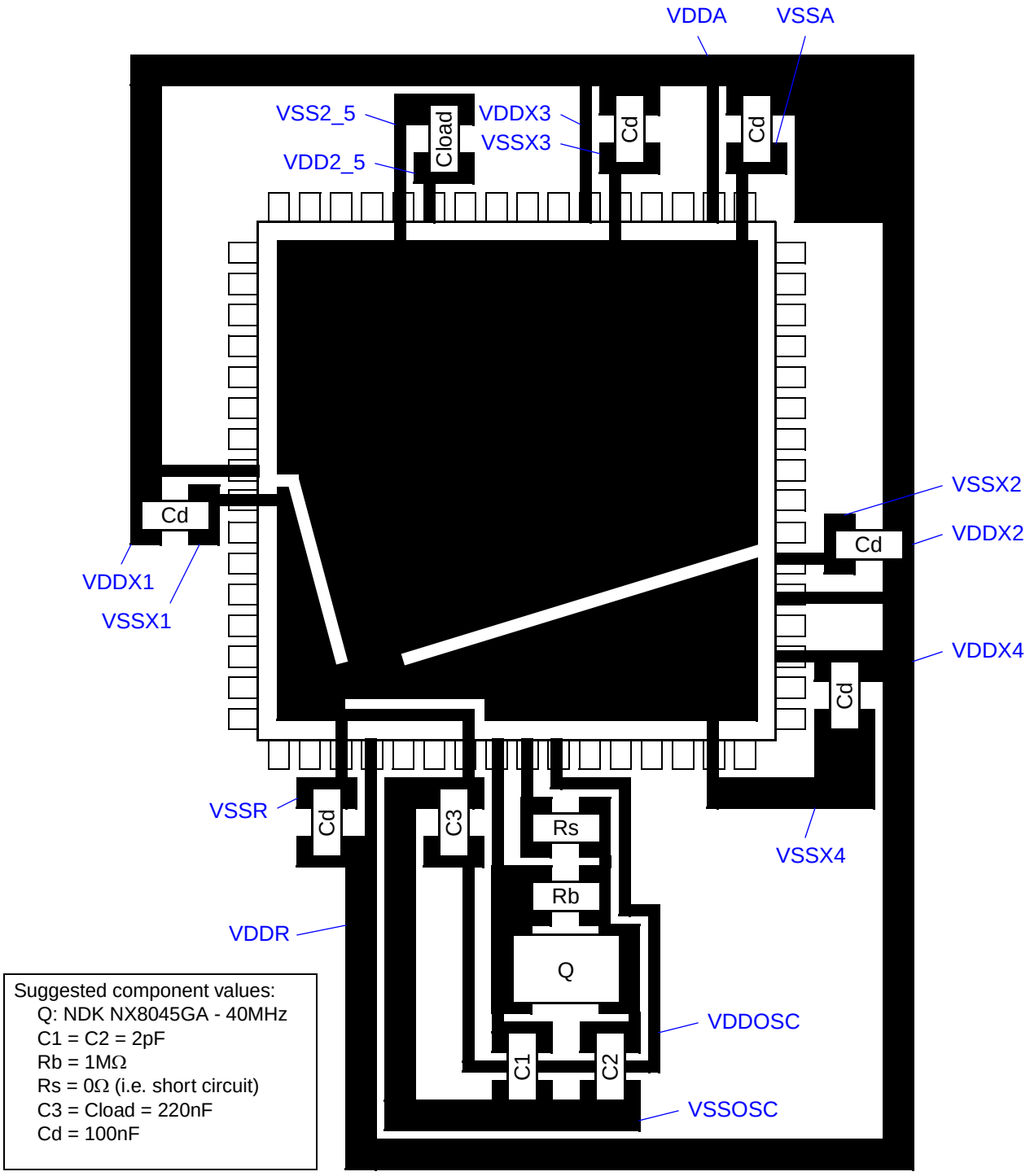


Figure C-1. Recommended PCB Layout (64-pin LQFP) for Standard Pierce Oscillator Mode

Appendix D
Index of Registers

N

O

P

R

S

T

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
+1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Document Number: MFR4310RM
Rev. 2
06/2007

freescale™
semiconductor