E-Ray

FlexRay IP Module

Application Note AN004

FIFO Function

Date: December 3rd, 2007

for IP Revision 1.0.2

APP_cover.fm

Robert Bosch GmbH
Automotive Electronics
Semiconductors and Integrated Circuits
Digital CMOS Design Group

Copyright Notice

Disclaimer

ROBERT BOSCH GMBH, MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

ROBERT BOSCH GMBH, RESERVES THE RIGHT TO MAKE CHANGES WITHOUT FURTHER NOTICE TO THE PRODUCTS DESCRIBED HEREIN. ROBERT BOSCH GMBH DOES NOT ASSUME ANY LIABILITY ARISING OUT OF THE APPLICATION OR USE OF ANY PRODUCT OR CIRCUIT DESCRIBED HEREIN.

1. About this Document 4

1.1 Change Control 4

1.1.1 Current Status 4

1.1.2 Change History 4

1.2 References 4

1.3 Terms and Abbreviations 4

2. Introduction 5

3. Version Control 5

4. Procedure 6

4.1 Overview 6

4.2 Functionality 7

4.3 FIFO Configuration Registers 8

4.3.1 FIFO Rejection Filter (FRF) 8

4.3.2 FIFO Rejection Filter Mask (FRFM) 9

4.3.2.1 FIFO Rejection Filter Configuration Example 9

4.3.3 FIFO Critical Level (FCL) 11

4.4 FIFO Message RAM 12

4.4.1 Configuration of MRC 12

4.4.2 Configuration of the FIFO 12

4.4.3 FIFO Reconfiguration in Normal Operation Mode 13

4.4.4 Access to the FIFO outside POC state CONFIG 13

4.5 Application Examples 15

4.5.1 Example One 15

4.5.1.1 FlexRay Channel Traffic 15

4.5.1.2 Message RAM Configuration 16

4.5.1.3 FIFO Configuration 17

4.5.1.4 FIFO Status Flags 18

4.5.1.5 Access to the FIFO during cycle 9 22

4.5.1.6 Empty FIFO Access 23

4.5.2 Example Two 24

4.5.2.1 FlexRay Channel Traffic 24

4.5.2.2 Message RAM Configuration 25

4.5.2.3 FIFO Configuration 26

4.5.2.4 FIFO Status Flags 27

4.5.2.5 Access to the FIFO during cycle 9 31

4.5.2.6 Empty FIFO Access 32

5. List of Tables 33

6. List of Figures 34

APP_TOC.fm

1. About this Document

1.1 Change Control

1.1.1 Current Status

Version 1.0.3

1.1.2 Change History

Issue	Date	By	Change
Version 1.0.0	19.05.2006	Kay Hammer	Initial Draft for IP Revision 1.0
Version 1.0.1	03.11.2006	Kay Hammer	Update for E-Ray Specification 1.2.3
Version 1.0.2	01.06.2007	Kay Hammer	Update for E-Ray Specification 1.2.5
Version 1.0.3	03.12.2007	Kay Hammer	Update for IP Revision 1.0.2

1.2 References

This document refers to the following documents:

Ref	Author(s)	Title
1	FlexRay Group	FlexRay Protocol Specification 2.1
2	Robert Bosch GmbH	E-Ray FlexRay IP-Module User’s Manual 1.2.6

1.3 Terms and Abbreviations

This document uses the following terms and abbreviations:

Term	Meaning
CC	Communication Controller
FIFO	First In First Out (message buffer structure)

APP_about.fm

2. Introduction

This application note describes the configuration of the FIFO message RAM and the FIFO function of the IP module.

The application note shall be used together with Ref. 1 and Ref. 2.

3. Version Control

This application note is based on the E-Ray FlexRay IP-module Revision 1.0.2. To verify the IP-module version, the Host can read out the Core Release Register (CREL). The register is read only. The correct value for the revision 1.0.2 is CREL = 0x10271031 (see Ref. 2).

4. Procedure

4.1 Overview

A group of the message buffers can be configured as a cyclic First-In-First-Out (FIFO) buffer. The group of message buffers belonging to the FIFO is contiguous in the register map starting with the message buffer referenced by **MRC.FFB[7:0]** and ending with the message buffer referenced by **MRC.LCB[7:0]**. Up to 127 message buffers can be assigned to the FIFO, at least message buffer 0 shall be reserved for transmission in the key slot due to FlexRay specification requirements.

Every **valid** incoming message not matching with any dedicated receive buffer but passing the programmable FIFO filter is stored into the FIFO. In this case frame ID, payload length received, receive cycle count, and the message buffer status MBS of the addressed FIFO message buffer are overwritten with frame ID, payload length, receive cycle count, and the status from the received frame. Bit **SIR.RFNE** shows that the FIFO is not empty, bit **SIR.RFCL** is set when the receive FIFO fill level **FSR.RFFL[7:0]** is equal or greater than the critical level as configured by **FCL.CL[7:0]**, bit **EIR.RFO** shows that a FIFO overrun has been detected.

If null frames are not rejected by the FIFO rejection filter, the null frames will be treated like data frames when they are stored into the FIFO.

APP_description.fm

4.2 Functionality

There are two index registers associated with the FIFO. The PUT Index Register (PIDX) is an index to the next available location in the FIFO. When a new message has been received it is written into the message buffer addressed by the PIDX register. The PIDX register is then incremented and addresses the next available message buffer. If the PIDX register is incremented past the highest numbered message buffer of the FIFO, the PIDX register is loaded with the number of the first (lowest numbered) message buffer in the FIFO chain. The GET Index Register (GIDX) is used to address the next message buffer of the FIFO to be read. The GIDX register is incremented after transfer of the contents of a message buffer belonging to the FIFO to the Output Buffer. The PUT Index Register and the GET Index Register are not accessible by the Host.

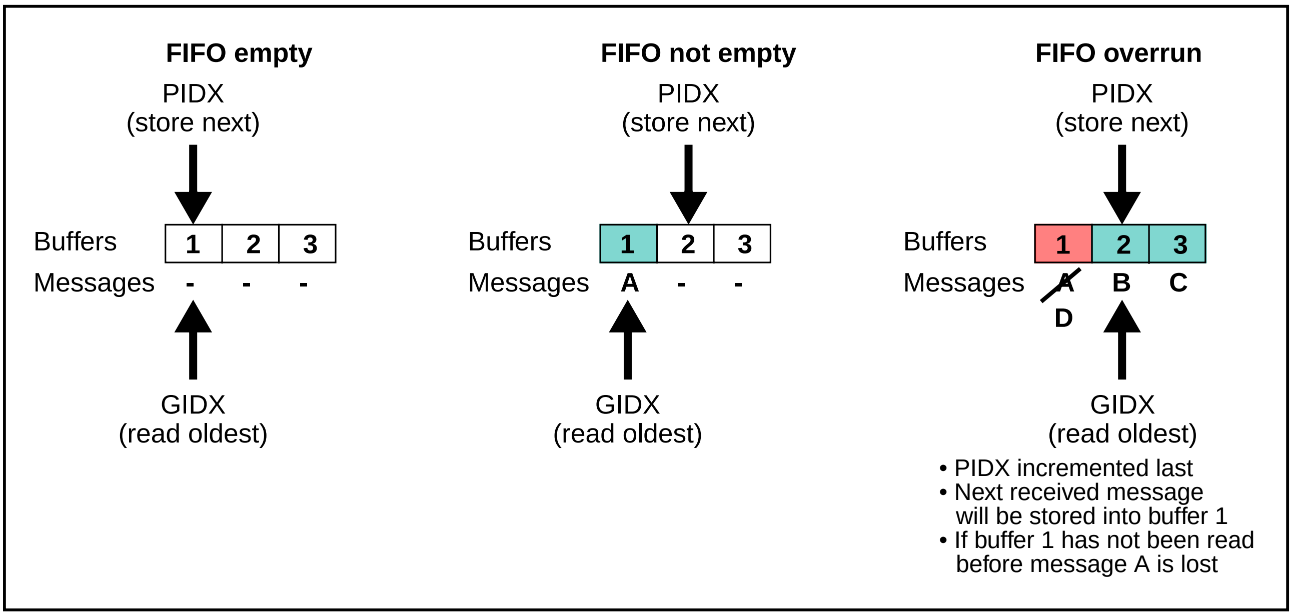


Figure 1: FIFO status: empty, not empty, overrun

The FIFO is completely filled when the PUT index (PIDX) reaches the value of the GET index (GIDX). When the next message is written to the FIFO before the oldest message has been read, both PUT index and GET index are incremented and the new message overwrites the oldest message in the FIFO. This will set FIFO overrun flag **EIR.RFO**.

A FIFO non empty status is detected when the PUT index (PIDX) differs from the GET index (GIDX) or when FIFO is full. In this cases flag **SIR.RFNE** is set. This indicates that there is at least one received message in the FIFO. The FIFO empty, FIFO not empty, and the FIFO overrun states are explained in figure 1 for a three message buffer FIFO.

4.3 FIFO Configuration Registers

4.3.1 FIFO Rejection Filter (FRF)

The programmable FIFO Rejection Filter (FRF) defines a filter pattern for messages to be rejected. The FIFO filter consists of channel filter, frame ID filter, and cycle counter filter. If bit **FRF.RSS** is set to ’1’ (default), all messages received in the static segment are rejected by the FIFO. If bit **FRF.RNF** is set to ’1’ (default), received null frames are not stored in the FIFO.

The FRF register can be written during POC state DEFAULT_CONFIG or CONFIG only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRF	R	0	0	0	0	0	0	0	RNF*	RSS*	CYF6*	CYF5*	CYF4*	CYF3*	CYF2*	CYF1*	CYF0*
0x0304	W																
Reset		0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit		15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	R	0	0	0	FID10*	FID9*	FID8*	FID7*	FID6*	FID5*	FID4*	FID3*	FID2*	FID1*	FID0*	CH1*	CH0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CH[1:0] Channel Filter

11 = no reception
10 = receive only on channel A
01 = receive only on channel B
00 = receive on both channels

Note: If reception on both channels is configured, also in static segment always both frames (from channel A and B) are stored in the FIFO, even if they are identical.

FID[10:0] Frame ID Filter

Determines the frame ID to be **rejected** by the FIFO. With the additional configuration of register **FRFM**, the corresponding frame ID filter bits are ignored, which results in further rejected frame IDs. When **FRFM.MFID[10:0]** is zero, a frame ID filter value **FRF.FID** = "0" means that no frame ID is **rejected**.

0...2047 = Frame ID filter values

CYF[6:0] Cycle Counter Filter

The 7-bit cycle counter filter determines the cycle set to which frame ID and channel rejection filter are applied. In cycles **not** belonging to the cycle set specified by **CYF[6:0]**, all frames are rejected.

RSS Reject in Static Segment

If this bit is set, the FIFO is used only for the dynamic segment.

1 = Reject messages in static segment
0 = FIFO also used for static segment

RNF Reject Null Frames

If this bit is set, received null frames are not stored in the FIFO.

1 = Reject all null frames
0 = Null frames are stored in the FIFO (treated like data frames)

4.3.2 FIFO Rejection Filter Mask (FRFM)

The FIFO Rejection Filter Mask specifies which of the corresponding frame ID filter bits are relevant for rejection filtering.

If a bit is set, it indicates that the corresponding bit in the FRF register will not be considered for rejection filtering. As a result, the FRFM specifies which bits of the frame ID filter (**FRF.FID[10:0]**) are marked as **don’t care** for rejection filtering.

The FRFM register can be written during POC state DEFAULT_CONFIG or CONFIG only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FRFM	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0308	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

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

MFID[10:0] Mask Frame ID Filter

1 = Ignore corresponding frame ID filter bit (marked as **don’t care**)

0 = Corresponding frame ID filter bit is used for rejection filtering

4.3.2.1 FIFO Rejection Filter Configuration Example

Together with the FIFO Rejection Filter register (FRF) the FIFO Rejection Filter Mask register (FRFM) determines whether a message with an assigned frame ID is rejected by the FIFO or not. The FRFM specifies which bits of the frame ID filter (**FRF.FID[10:0]**) are marked as **don’t care** for rejection filtering.

Rejection Filter Example 1:

FRF.FID =	"000 0000 0011"
FRFM.MFID =	"000 0000 0000"
rejected frame ID =	"000 0000 0011"

Bits **FRF.FID[10:0]** are fixed for determining the rejected frame IDs. Therefore, only frame ID 3 is rejected.

Configuration Example 2:

FRF.FID =	"000 0000 0011"
FRFM.MFID =	"000 0000 0001"
rejected frame IDs =	"000 0000 001-"

Bits **FRF.FID[10:1]** are fixed for determining the rejected frame IDs. Bit **FRF.FID[0]** is don’t care. Therefore, frame ID 2 and 3 are rejected.

Configuration Example 3:

FRF.FID = "000 0000 1000"
FRFM.MFID = "000 0000 0111"
rejected frame IDs = "000 0000 1---"

Bits **FRF.FID[10:3]** are fixed for determining the rejected frame IDs. Bits **FRF.FID[2:0]** are don’t care. Therefore, frame ID 8 up to 15 are rejected.

Configuration Example 4:

FRF.FID = "000 0000 1000"
FRFM.MFID = "000 0000 0100"
rejected frame IDs = "000 0000 1-00"

Bits **FRF.FID[10:4]** and **FRF.FID[1:0]** are fixed for determining the rejected frame IDs. Bit **FRF.FID[2]** is don’t care. Therefore, frame ID 8 and 12 are rejected.

Configuration Example 5:

FRF.FID = "000 0000 1000"
FRFM.MFID = "000 0001 1111"
rejected frame IDs = "000 000- ----"

Bits **FRF.FID[10:5]** are fixed for determining the rejected frame IDs. Bits **FRF.FID[4:0]** are don’t care. Therefore, frame ID 1 up to 31 are rejected.

4.3.3 FIFO Critical Level (FCL)

The CC accepts modifications of the register in POC state DEFAULT_CONFIG or CONFIG only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FCL	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x030C	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit		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	CL7*	CL6*	CL5*	CL4*	CL3*	CL2*	CL1*	CL0*
	W																
Reset		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

CL[7:0] Critical Level

When the receive FIFO fill level **FSR.RFFL[7:0]** is equal or greater than the critical level configured by **CL[7:0]**, the receive FIFO critical level flag **FSR.RFCL** is set. If **CL[7:0]** is programmed to values > 128, bit **FSR.RFCL** is never set. When **FSR.RFCL** changes from ’0’ to ’1’ bit **SIR.RFCL** is set to ’1’, and if enabled, an interrupt is generated.

4.4 FIFO Message RAM

4.4.1 Configuration of MRC

The Message RAM Configuration Register (MRC) defines the number of message buffers assigned to the static segment, dynamic segment and FIFO.

The FIFO size is configured via **MRC.FFB** and **MRC.LCB**.

If a dynamic buffer and a FIFO are configured, **MRC.FFB** has to be greater than **MRC.FDB**. It is not possible to place the FIFO header section in front of the dynamic header section.

To disable dynamic buffers, **MRC.FDB** has to be programmed to a value greater than or equal to 128.

To disable the FIFO, **MRC.FFB** has to be programmed to a value greater than or equal to 128.

If dynamic and/or FIFO message buffers are configured, **MRC.LCB** must be always greater than **MRC.FFB** and **MRC.FDB**. It is not possible to place the FIFO header section or the dynamic header section in front of the static header section.

The FlexRay protocol specification requires that each node has to send a frame in its key slot. Therefore at least message buffer 0 is reserved for transmission in key slot. Due to this requirement a maximum number of 127 message buffers can be assigned to the FIFO.

The programmer has to ensure that the configuration defined by **FDB[7:0]**, **FFB[7:0]** and **LCB[7:0]** is valid. **The CC does not check for erroneous configurations!**

4.4.2 Configuration of the FIFO

The FIFO size is configured via **MRC.FFB** and **MRC.LCB**. A maximum number of 127 message buffers can be assigned to the FIFO.

Each message buffer of the FIFO has to be configured individually with the payload length **WRHS2.PLC[6:0]** and the data pointer **WRHS3.DP[10:0]**. The payload length configured must be identical for all FIFO buffers.

The header section of each FIFO buffer is configured after the following rules:

- All bits of WRHS1 shall be configured to "0". All informations required by the CC for acceptance filtering is taken from the FIFO rejection filter (FRF) and the FIFO rejection filter mask (FRFM).
- In WRHS2, the payload length configured of the FIFO buffer has to be configured (**WRHS2.PLC[6:0]**). The payload length configured must be identical for all FIFO buffers. The configuration of **WRHS2.CRC[10:0]** is not necessary.
- In WRHS3, the data pointer according to the FIFO buffer has to be configured. The application programmer has to ensure that the data sections of the different FIFO buffers are not overlapping.

Each FIFO buffer must be configured individually with the following Host access sequence:

- Write header section to WRHS1..3
- Write Command Mask: write **IBCM.LDSH**
- Wait until **IBCR.IBSYH** is reset
- Demand data transfer to target message buffer: write **IBCR.IBRH[6:0]**

4.4.3 FIFO Reconfiguration in Normal Operation Mode

If a FIFO message buffers needs to be reconfigured outside POC state DEFAULT_CONFIG or CONFIG (e.g. because of a parity error in the locked FIFO header section), the locked header section has to be unlocked. For such a transfer, the write-access to the IBCR (specifying the message buffer number) must be immediately preceded by the unlock sequence normally used to leave CONFIG state (see Ref. 2, chapter 4.3.3 Lock Register (LCK)).

For that single transfer the respective message buffer header is unlocked, regardless whether it belongs to the FIFO or whether its locking is controlled by **MRC.SEC[1:0]**, and will be updated with new data.

4.4.4 Access to the FIFO outside POC state CONFIG

To access the messages stored in the FIFO outside POC state DEFAULT_CONFIG or CONFIG, the Host has to trigger a transfer from the Message RAM to the Output Buffer by writing the number of the first message buffer of the FIFO (referenced by **MRC.FFB[7:0]**) to the register OBCR.

The Message Handler then transfers the message buffer addressed by the GET Index Register (GIDX) to the Output Buffer. After this transfer the GET Index Register (GIDX) is incremented (see chapter 4.2).

Note: To read out the messages stored in the FIFO, the Host must trigger each transfer from the Message RAM to the Output Buffer by writing always the number of the first message buffer of the FIFO (referenced by **MRC.FFB[7:0]**) to **OBCR.REQ**.

The FIFO buffer has to be read out with the following Host access sequence:

- Write Output Buffer Command Mask **OBCM.RHSS**, **OBCM.RDSS**
- Wait until **OBCR.OBSYS** is reset
- Request transfer of first FIFO message buffer to OBF Shadow by writing **OBCR.OBRS[6:0]** and **OBCR.REQ** (in case of and 8-bit Host interface, **OBCR.OBRS[6:0]** has to be written before **OBCR.REQ**). **OBCR.REQ** must be always the first message buffer of the FIFO (referenced by **MRC.FFB[7:0]**).
- Wait until **OBCR.OBSYS** is reset
- Toggle OBF Shadow and OBF Host by writing **OBCR.VIEW** = "1"
- Read out transferred message buffer by reading RDDSn, RDHS1..3 and MBS.

Note: The RDHS1 register only holds the frame ID of the received frame (**RDHS1.FID[10:0]**). All other bits are read as "0".

Note: If the Host triggers the CC to transfer the data section of the FIFO buffer from the Message RAM to the Output Buffer (**OBCM.RDSS** = "1"), always the **configured** payload length of the FIFO buffer is transferred (**WRHS2.PLC[6:0]**, **RDHS2.PLC[6:0]**, respectively), and not the payload length received and stored into the message buffer (**RDHS2.PLR[6:0]**). If the payload length received is smaller than the payload length configured, the remaining data bytes of the data section are filled with undefined data.

Note: When a message is stored into the FIFO buffer the following behaviour with respect to payload length received (**RDHS2.PLR[6:0]**) and payload length configured (**WRHS2.PLC[6:0]**, **RDHS2.PLC[6:0]**, respectively) is implemented:

PLR[6:0] > **PLC[6:0]**: The payload data stored in the message buffer is truncated to the payload length configured if **PLC[6:0]** even or else truncated to **PLC[6:0]** + 1.

PLR[6:0] ≤ **PLC[6:0]**: The received payload data is stored into the message buffers data section. The remaining data bytes of the data section as configured by **PLC[6:0]** are filled with undefined data

PLR[6:0] = zero: The message buffer’s data section is filled with undefined data

PLC[6:0] = zero: Message buffer has no data section configured. No data is stored into the message buffer’s data section.

4.5 Application Examples

In this chapter, the function of the FIFO is illustrated on the basis of different communication cycles and node configurations. Wakeup and Startup are not part of the example, rather the node is in POC state NORMAL_ACTIVE (normal operation mode). Only the message handling, depending on the FlexRay channel traffic, is described.

4.5.1 Example One

4.5.1.1 FlexRay Channel Traffic

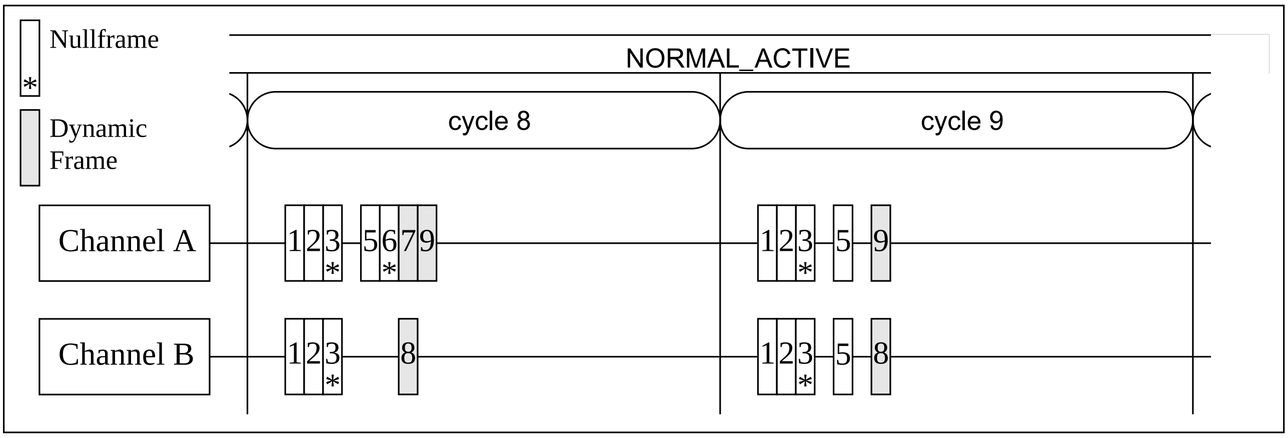


Figure 2: FlexRay channel traffic example one

On the basis of the two communication cycles shown in figure 2 the function of the FIFO is described.

Frame ID 1 up to 6 are static frames, with frame ID 3 and 6 configured as Nullframes. Frame ID 1 is configured as transmit frame (see chapter 4.5.1.2).

Frame ID 7 up to 9 are dynamic frames.

All frames are valid frames.

4.5.1.2 Message RAM Configuration

The number of message buffers in this example is 8. Therefore, **MRC.LCB** = 7. All data pointers must be greater than or equal to (**MRC.LCB** + 1) * 4 = 32.

The header partition contains 3 static message buffers and 5 FIFO buffers. The FIFO size is configured by programming **MRC.FFB** = 3 and **MRC.LCB** = 7. Dynamic message buffers are disabled by programming **MRC.FDB** = 128 (MRC = 0x00070380).

The configured static frame data length is **MHDC.SFDL[6:0]** = 4 (two byte words). The configured payload for each FIFO message buffer is also 4 (two byte words).

The Message RAM is configured as follows:

RAM Word					
0..3	Message Buffer 0	TX, FID = 1, DP = 0x20			Start of Header Partition
4..7	Message Buffer 1	RX, FID = 2, DP = 0x22			
8..11	Message Buffer 2	RX, FID = 3, DP = 0x24			
12..15	Message Buffer 3	FIFO, DP = 0x26		⇐ **FFB**	
16..19	Message Buffer 4	FIFO, DP = 0x28			
20..23	Message Buffer 5	FIFO, DP = 0x2A			
24..27	Message Buffer 6	FIFO, DP = 0x2C			
28..31	Message Buffer 7	FIFO, DP = 0x2E		⇐ **LCB**	End of Header Partition
32		MB0 Data1	MB0 Data0	⇐ **DP MB0**	Start of Data Partition
33		MB0 Data3	MB0 Data2		
34		MB1 Data1	MB1 Data0	⇐ **DP MB1**	
35		MB1 Data3	MB1 Data2		
36		MB2 Data2	MB2 Data0	⇐ **DP MB2**	
37		MB2 Data3	MB2 Data2		
38		MB3 Data1	MB3 Data0	⇐ **DP MB3**	
39		MB3 Data3	MB3 Data2		
40		MB4 Data1	MB4 Data0	⇐ **DP MB4**	
41		MB4 Data3	MB4 Data2		
42		MB5 Data1	MB5 Data0	⇐ **DP MB5**	
43		MB5 Data3	MB5 Data2		
44		MB6 Data1	MB6 Data0	⇐ **DP MB6**	
45		MB6 Data3	MB6 Data2		
46		MB7 Data1	MB7 Data0	⇐ **DP MB7**	
47		MB7 Data3	MB7 Data2		
48..2046		unused			
2047		unused			End of Data Partition

Figure 3: Message RAM structure for application example one

4.5.1.3 FIFO Configuration

The Message RAM is configured as described in chapter 4.5.1.2.

The FIFO Rejection Filter (FRF) is configured with the following values:

FRF.CH[1:0]	= "00"	(Receive on both channels)
FRF.FID[10:0]	= "0"	(No frame ID is rejected)
FRF.CYF[6:0]	= "0"	(Frame ID and channel rejection filter are applied to all cycles)
FRF.RSS	= "0"	(FIFO used also for static segment)
FRF.RNF	= "0"	(Null frames are stored in static segment)

As a result, FRF = 0x00000000.

The FIFO Rejection Filter Mask (FRFM) is configured with the following value:

FRFM.MFID[10:0] = "0" (No bits are marked as **don’t care** for rejection filtering)

As a result, FRFM = 0x00000000.

The FIFO Critical Level (FCL) is configured with the following value:

FCL.CL[7:0] = "0000 0011" (FIFO critical level is 3)

As a result, FCL = 0x00000003.

4.5.1.4 FIFO Status Flags

The FIFO is empty at the end of cycle 7.

In cycle 8,

- slot one, the frame ID 1 (key slot frame) is transmitted by the FIFO node,
- slot two, the frame ID 2 matches the receive buffer, message buffer one, and is stored into it,
- slot three, the frame ID 3 matches the receive buffer, message buffer two, and is stored into it,
- slot four, no traffic occurs on the FlexRay channel,
- slot five, the frame ID 5 does not matching any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flags change:
  - **EIR.RFNE** = "1" (Receive FIFO Not Empty)
  - **FSR.RFNE** = "1" (Receive FIFO Not Empty)
  - **FSR.RFFL[7:0]** = "00000001" (Receive FIFO Fill Level)
- slot six, the frame ID 6 does not matching any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flags change:
  - **FSR.RFFL[7:0]** = "00000010" (Receive FIFO Fill Level)
- slot seven (dynamic segment), the frame ID 7 does not matching any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flags change:
  - **SIR.RFCL** = "1" (Receive FIFO Critical Level), because the FIFO critical level was configured to 3 (**FCL.CL[7:0]** = "0011")
  - **FSR.RFCL** = "1" (Receive FIFO Critical Level), because the FIFO critical level was configured to 3 (**FCL.CL[7:0]** = "0011")
  - **FSR.RFFL** = "00000011" (Receive FIFO Fill Level)
- slot eight (dynamic segment), the frame ID 8 does not matching any receive buffer. The frame occurs on channel B. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flag change:
  - **FSR.RFFL** = "00000100" (Receive FIFO Fill Level)
- slot nine (dynamic segment), the frame ID 9 does not matching any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flag change:
  - **FSR.RFFL** = "00000101" (Receive FIFO Fill Level)

The values of the FIFO related status flags after slot 9 in cycle 8 are:

- **EIR.RFO** = "0" (Receive FIFO Overrun)
- **EIR.EFA** = "0" (Empty FIFO Access)
- **SIR.RFNE** = "1" (Receive FIFO Not Empty)
- **SIR.RFCL** = "1" (Receive FIFO Critical Level)
- **FSR.RFNE** = "1" (Receive FIFO Not Empty)
- **FSR.RFCL** = "1" (Receive FIFO Critical Level)
- **FSR.RFO** = "0" (Receive FIFO Overrun)
- **FSR.RFFL[7:0]** = "00000101" (Receive FIFO Fill Level)

Message Buffer 3 / FIFO Buffer 1	Frame ID 5, channel A, from cycle 8
Message Buffer 4 / FIFO Buffer 2	Frame ID 6, channel A, from cycle 8
Message Buffer 5 / FIFO Buffer 3	Frame ID 7, channel A, from cycle 8
Message Buffer 6 / FIFO Buffer 4	Frame ID 8, channel B, from cycle 8
Message Buffer 7 / FIFO Buffer 5	Frame ID 9, channel A, from cycle 8

Figure 4: Content of FIFO message buffers at the end of cycle 8, example one

In cycle 9,

- slot one, the frame ID 1 (key slot frame) is transmitted by the FIFO node,
- slot two, the frame ID 2 matches the receive buffer, message buffer one, and is stored into it,
- slot three, the frame ID 3 matches the receive buffer, message buffer two, and is stored into it,
- slot four, no traffic occurs on the FlexRay channel,
- slot five, the frame ID 5 does not matching any receive buffer. The frame occurs on both channels A and B. The frame is not rejected by the programmed FIFO filter. Both are stored into the FIFO, because no channel dependent rejection is configured (**FRF.CH[1:0]** = "00"). The two oldest FIFO message buffers (in that case, message buffer 3 and 4) were overwritten by theses two frames with frame ID 5. The old messages got lost. Hence, the following status flags change:
  - **EIR.RFO** = "1" (Receive FIFO Overrun)
  - **FSR.RFO** = "1" (Receive FIFO Overrun)

Note: It is not predictability which from the both frames from channel A or B with frame ID 5 are stored at first into the FIFO.

Note: The Receive FIFO Fill Level reminds on the maximum fill level (**FSR.RFFL[7:0]** = "00000101").

- slot six, no traffic occurs on the FlexRay channel,
- slot seven (dynamic segment), no traffic occurs on the FlexRay channel,
- slot eight (dynamic segment), the frame ID 8 does not matching any receive buffer. The frame occurs on channel B. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. The oldest FIFO message buffer (in that case, message buffer 5) is overwritten.The old message got lost.
- slot nine (dynamic segment), the frame ID 9 does not matching any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. The oldest FIFO message buffer (in that case, message buffer 6) is overwritten. The old message got lost.

The values of the FIFO related status flags after slot 9 in cycle 9 are:

• **EIR.RFO**	= "1"	(Receive FIFO Overrun)
• **EIR.EFA**	= "0"	(Empty FIFO Access)
• **SIR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **SIR.RFCL**	= "1"	(Receive FIFO Critical Level)
• **FSR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **FSR.RFCL**	= "1"	(Receive FIFO Critical Level)
• **FSR.RFO**	= "1"	(Receive FIFO Overrun)
• **FSR.RFFL[7:0]**	= "00000101"	(Receive FIFO Fill Level)

Message Buffer 3 / FIFO Buffer 1	Frame ID 5, channel A (or B)*, from cycle 9
Message Buffer 4 / FIFO Buffer 2	Frame ID 5, channel B (or A)*, from cycle 9
Message Buffer 5 / FIFO Buffer 3	Frame ID 8, channel B, from cycle 9
Message Buffer 6 / FIFO Buffer 4	Frame ID 9, channel A, from cycle 9
Message Buffer 7 / FIFO Buffer 5	Frame ID 9, channel A, from cycle 8

* It is not predictability which from the both frames from channel A and B are stored at first into the FIFO.

Figure 5: Content of FIFO message buffers at the end of cycle 9, example one

4.5.1.5 Access to the FIFO during cycle 9

In cycle 9, after slot 9, the Host reads out the FIFO as described in chapter 4.4.3. Depending from the configuration of **OBCM.RDSS** and **OBCM.RHSS**, data and/or header section are transferred from the Message RAM to the Output Buffer.

If the Host triggers the CC to transfer the data section of the FIFO buffer from the Message RAM to the Output Buffer (**OBCM.RDSS** = "1"), always the **configured** payload length of the FIFO buffer is transferred (**WRHS2.PLC[6:0]**, **RDHS2.PLC[6:0]**, respectively), and not the payload length received and stored into the message buffer (**RDHS2.PLR[6:0]**). If the payload length received is smaller than the payload length configured, the remaining data bytes of the data section are filled with undefined data.

To access the FIFO, always the first FIFO buffer, referenced by **MRC.FFB[7:0]**, must be read out. In this example, the first FIFO buffer which must be always read out by the Host is buffer 3 (**MRC.FFB** = 3).

```
/* write32bit(address, value) */
write32bit(0x0710, 0x00000003);   /* Write Output Buffer Command Mask
                                     OBCM.RHSS, OBCM.RDSS */

while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000203);   /* Request transfer of first FIFO message
                                     buffer MRC.FFB[7:0] to OBF Shadow by
                                     writing OBCR.OBRS[6:0] and OBCR.REQ */
```

Note: After the "Request Message RAM Transfer" command (**OBCR.REQ** = "1", **OBCR.OBRS** = "11"), the following FIFO related status flags change (see table 1):

- **EIR.RFO** = "0" (Receive FIFO Overrun)
- **FSR.RFO** = "0" (Receive FIFO Overrun)
- **FSR.RFFL[7:0]** = "00000100" (Receive FIFO Fill Level)

```
while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000303);   /* Toggle OBF Shadow and OBF Host by
                                     writing OBCR.VIEW = "1".
                                     Request transfer of first FIFO message
                                     buffer MRC.FFB[7:0] to OBF Shadow by
                                     writing OBCR.OBRS[6:0] and OBCR.REQ */

value32bit = read32bit(0x0700);   /* Read out transferred message buffer
                                     by reading RDHS1 (and RDHS2..3,
                                     RDDSn, MBS). The RDHS1 register
                                     only holds the frame
                                     ID of the received frame
                                     (RDHS1.FID[10:0]), all other bits
                                     are read as "0" */
```

After each of these transfer requests, FIFO related status flags change as described in the following table (**n** is the point in time when the Host triggers the CC for the first transfer from the Message RAM to the Output Buffer).

	Request Message RAM Transfer	FIFO related status flags after the "Request Message RAM Transfer" command								
Time	**OBCR**	**EIR. RFO**	**EIR. EFA**	**SIR. RFNE**	**SIR. RFCL**	**FSR. RFNE**	**FSR. RFCL**	**FSR. RFO**	**FSR. RFFL**	**RDHS1. FID**
n-1	-	1	0	1	1	1	1	1	5	-
n	`0x00000203`	0	0	1	1	1	1	0	4	-
n+1	`0x00000303`	0	0	1	1	1	1	0	3	9 (cycle 8)
n+2	`0x00000303`	0	0	1	0	1	0	0	2	5 (cycle 9)
n+3	`0x00000303`	0	0	1	0	1	0	0	1	5 (cycle 9)
n+4	`0x00000303`	0	0	0	0	0	0	0	0	8 (cycle 9)
n+5	`0x00000103`	0	0	0	0	0	0	0	0	9 (cycle 9)

Table 1: FIFO status flags during read out, example one

4.5.1.6 Empty FIFO Access

After the Host has triggered the fifth transfer from the Message RAM to the Output Buffer, the FIFO is empty (**FSR.RFFL[7:0]** = "00000000", see table 1.) If after that point in time the Host triggers another transfer from the Message RAM to the Output Buffer from the FIFO, the flag "Empty FIFO Access" is set (**EIR.EFA** = "1").

```
value32bit = read32bit(0x0318);
```

Note: The value of the Receive FIFO Fill Level is **FSR.RFFL[7:0]** = "00000000"

```
write32bit(0x0710, 0x00000003);    /* Write Output Buffer Command Mask
                                      OBCM.RHSS, OBCM.RDSS */

while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000203);    /* Request transfer of first FIFO message
                                      buffer MRC.FFB[7:0] to OBF Shadow by
                                      writing OBCR.OBRS[6:0] and OBCR.REQ */
```

Note: After the Host has triggered the transfer from the Message RAM to the Output Buffer from the empty FIFO, the flag **EIR.EFA** is set to "1".

4.5.2 Example Two

4.5.2.1 FlexRay Channel Traffic

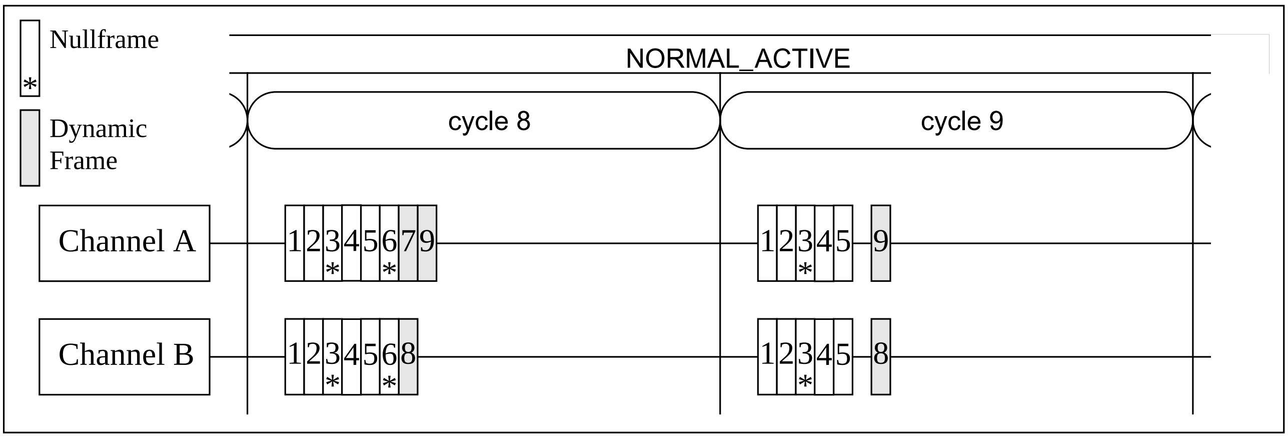


Figure 6: FlexRay channel traffic example two

On the basis of the two communication cycles shown in figure 6 the function of the FIFO is described.

Frame ID 1 up to 6 are static frames, with frame ID 3 and 6 configured as Nullframes.

Frame ID 7 up to 9 are dynamic frames.

All frames are valid messages.

4.5.2.2 Message RAM Configuration

The number of message buffers in this example is 9. Therefore, **MRC.LCB** = 8. All data pointers must be greater than or equal to (**MRC.LCB** + 1) * 4 = 36.

The header partition contains only FIFO buffers. The FIFO size is configured by programming **MRC.FFB** = 1 and **MRC.LCB** = 8. Dynamic message buffers are disabled by programming **MRC.FDB** = 128 (MRC = 0x00080180).

The configured payload for each FIFO message buffer is 4 (two byte words).

The Message RAM is configured as follows:

RAM Word					
0..3	Message Buffer 0	TX, FID = 4, DP = 0x24			Start of Header Partition
4..7	Message Buffer 1	FIFO, DP = 0x26		⇐ **FFB**	
8..11	Message Buffer 2	FIFO, DP = 0x28			
12..15	Message Buffer 3	FIFO, DP = 0x2A			
16..19	Message Buffer 4	FIFO, DP = 0x2C			
20..23	Message Buffer 5	FIFO, DP = 0x2E			
24..27	Message Buffer 6	FIFO, DP = 0x30			
28..31	Message Buffer 7	FIFO, DP = 0x32			
32..35	Message Buffer 8	FIFO, DP = 0x34		⇐ **LCB**	End of Header Partition
36		MB0 Data1	MB0 Data0	⇐ **DP MB0**	Start of Data Partition
37		MB0 Data3	MB0 Data2		
38		MB1 Data1	MB1 Data0	⇐ **DP MB1**	
39		MB1 Data 3	MB1 Data2		
40		MB2 Data1	MB2 Data0	⇐ **DP MB2**	
41		MB2 Data 3	MB2 Data2		
42		MB3 Data1	MB3 Data0	⇐ **DP MB3**	
43		MB3 Data3	MB3 Data2		
44		MB4 Data1	MB4 Data0	⇐ **DP MB4**	
45		MB4 Data3	MB4 Data2		
46		MB5 Data1	MB5 Data0	⇐ **DP MB5**	
47		MB5 Data3	MB5 Data2		
48		MB6 Data1	MB6 Data0	⇐ **DP MB6**	
49		MB6 Data3	MB6 Data2		
50		MB7 Data1	MB7 Data0	⇐ **DP MB7**	
51		MB7 Data3	MB7 Data2		
52		MB8 Data1	MB8 Data0	⇐ **DP MB8**	
53		MB7 Data8	MB8 Data2		
54..2046		unused			
2047		unused			End of Data Partition

Figure 7: Message RAM structure for application example two

4.5.2.3 FIFO Configuration

The Message RAM is configured as described in chapter 4.5.2.1.

The FIFO Rejection Filter (FRF) is configured with the following values:

FRF.CH[1:0] = "10" (Receive only on channel A)
FRF.FID[10:0] = "000 0000 1001" (Frame ID 9 is rejected)
FRF.CYF[6:0] = "0" (Frame ID and channel rejection filter are applied to all cycles)
FRF.RSS = "0" (FIFO used also for static segment)
FRF.RNF = "1" (Null frames are rejected)

As a result, FRF = 0x01000026.

The FIFO Rejection Filter Mask (FRFM) is configured with the following value:

FRFM.MFID[10:0] = "000 0000 1010"

Thus, bit 3 and bit 1 are marked as "don’t care" for rejection filtering.
As a result, FRFM = 0x00000028.

The rejection filter configuration leads to the following rejected frame IDs:

FRF.FID = `"000 0000 1001"`
FRFM.MFID = `"000 0000 1010"`
rejected frame ID = `"000 0000 -0-1"`

Bits **FRF.FID[10:4]**, **FRF.FID[2]** and **FRF.FID[0]** are fixed for determining the rejected frame IDs. Bit **FRF.FID[3]** and **FRF.FID[1]** are don’t care. Therefore, frame ID 1, 3, 9 and 11 are rejected.

The FIFO Critical Level (FCL) is configured with the following value:

FCL.CL[7:0] = "0000 0101" (FIFO critical level is 5)

As a result, FCL = 0x00000005.

4.5.2.4 FIFO Status Flags

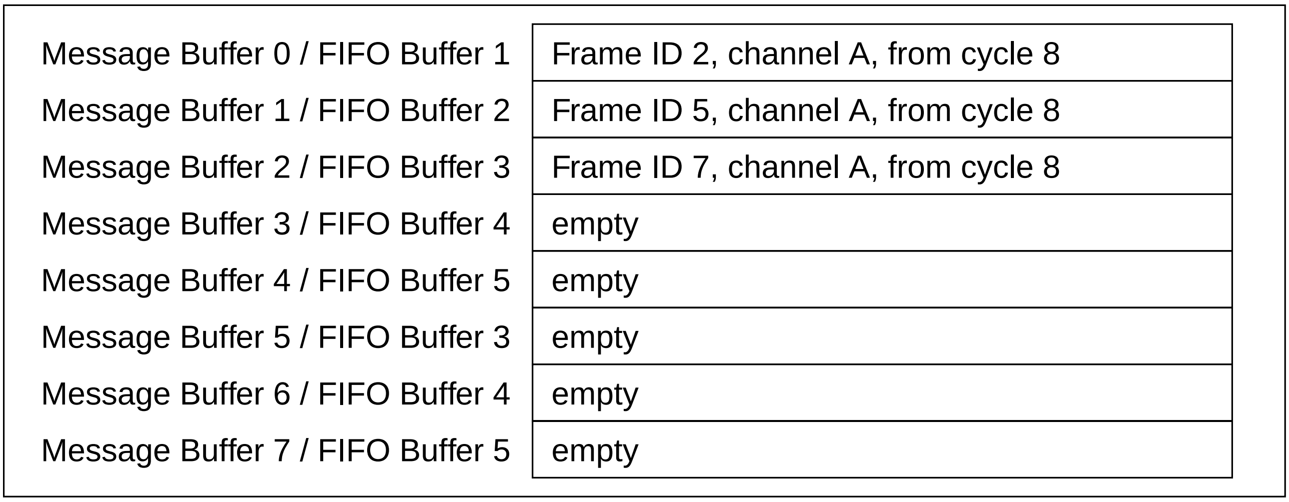
The FIFO is empty at the end of cycle 7.

In cycle 8,

- slot one, the frame ID 1 does not match any receive buffer. The frame occurs on both channel A and B. The frame is rejected by the FIFO filter (Frame ID 1 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM.)
- slot two, the frame ID 2 does not match any receive buffer. The frame occurs on both channel A and B. The frame from channel A is stored into the FIFO. The frame from channel B is rejected by the FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A). Hence, the following status flags change:
    - **EIR.RFNE** = "1" (Receive FIFO Not Empty)
    - **FSR.RFNE** = "1" (Receive FIFO Not Empty)
    - **FSR.RFFL** = "00000001" (Receive FIFO Fill Level)
- slot three, the frame ID 3 does not match any receive buffer. The frame occurs on both channel A and B. The frame is rejected by the FIFO filter (Frame ID 3 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM. Furthermore, the frame with frame ID 3 is a Nullframe, which is also rejected by the FIFO filter (**FRF.RNF** = "1")).
- slot four, the frame ID 4 (key slot frame) is transmitted by the FIFO node,
- slot five, the frame ID5 does not match any receive buffer. The frame occurs on both channel A and B. The frame from channel A is stored into the FIFO. The frame from channel B is rejected by the FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A). Hence, the following status flags change:
    - **FSR.RFFL** = "00000010" (Receive FIFO Fill Level)
- slot six, the frame ID 6 does not match any receive buffer. The frame occurs on both channel A and B. The frame is rejected by the FIFO filter (The frame with frame ID 6 is a Nullframe, which is rejected by the FIFO filter (**FRF.RNF** = "1")).
- slot seven (dynamic segment), the frame ID 7 does not match any receive buffer. The frame occurs on channel A. The frame is not rejected by the programmed FIFO filter. The frame is stored into the FIFO. Hence, the following status flags change:
    - **FSR.RFFL** = "00000011" (Receive FIFO Fill Level)
- slot eight (dynamic segment), the frame ID 8 does not match any receive buffer. The frame occurs on channel B. The frame is rejected by the programmed FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A).
- slot nine (dynamic segment), the frame ID 9 does not match any receive buffer. The frame occurs on channel A. The frame is rejected by the programmed FIFO filter (Frame ID 9 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM).

The values of the FIFO related status flags after slot 9 in cycle 8 are:

• **EIR.RFO**	= "0"	(Receive FIFO Overrun)
• **EIR.EFA**	= "0"	(Empty FIFO Access)
• **SIR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **SIR.RFCL**	= "0"	(Receive FIFO Critical Level)
• **FSR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **FSR.RFCL**	= "0"	(Receive FIFO Critical Level)
• **FSR.RFO**	= "0"	(Receive FIFO Overrun)
• **FSR.RFFL[7:0]**	= "00000011"	(Receive FIFO Fill Level)



Message Buffer 0 / FIFO Buffer 1	Frame ID 2, channel A, from cycle 8
Message Buffer 1 / FIFO Buffer 2	Frame ID 5, channel A, from cycle 8
Message Buffer 2 / FIFO Buffer 3	Frame ID 7, channel A, from cycle 8
Message Buffer 3 / FIFO Buffer 4	empty
Message Buffer 4 / FIFO Buffer 5	empty
Message Buffer 5 / FIFO Buffer 3	empty
Message Buffer 6 / FIFO Buffer 4	empty
Message Buffer 7 / FIFO Buffer 5	empty

Figure 8: Content of FIFO message buffers at the end of cycle 8, example two

In cycle 9,

- slot one, the frame ID 1 does not match any receive buffer. The frame occurs on both channel A and B. The frame is rejected by the FIFO filter (Frame ID 1 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM.)
- slot two, the frame ID 2 does not match any receive buffer. The frame occurs on both channel A and B. The frame from channel A is stored into the FIFO. The frame from channel B is rejected by the FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A). Hence, the following status flags change:
  - **FSR.RFFL** = "00000100" (Receive FIFO Fill Level)
- slot three, the frame ID 3 does not match any receive buffer. The frame occurs on both channel A and B. The frame is rejected by the FIFO filter (Frame ID 3 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM. Furthermore, the frame with frame ID 3 is a Nullframe, which is also rejected by the FIFO filter (**FRF.RNF** = "1")).
- slot four, the frame ID 4 (key slot frame) is transmitted by the FIFO node,
- slot five, the frame ID5 does not match any receive buffer. The frame occurs on both channel A and B. The frame from channel A is stored into the FIFO. The frame from channel B is rejected by the FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A). Hence, the following status flags change:
  - **SIR.RFCL** = "1" (Receive FIFO Critical Level)
  - **FSR.RFCL** = "1" (Receive FIFO Critical Level)
  - **FSR.RFFL[7:0]** = "00000101" (Receive FIFO Fill Level)
- slot six, no traffic occurs on the FlexRay channels,
- slot seven (dynamic segment), no traffic occurs on the FlexRay channels,
- slot eight (dynamic segment), the frame ID 8 does not match any receive buffer. The frame occurs on channel B. The frame is rejected by the programmed FIFO filter (**FRF.CHA[1:0]** = "10" receive only on channel A).
- slot nine (dynamic segment), the frame ID 9 does not match any receive buffer. The frame occurs on channel A. The frame is rejected by the programmed FIFO filter (Frame ID 9 belongs to the rejected frame IDs, configured by FIFO Rejection Filter FRF and FIFO Rejection Filter Mask FRFM.)

The values of the FIFO related status flags after slot 9 in cycle 9 are:

• **EIR.RFO**	= "0"	(Receive FIFO Overrun)
• **EIR.EFA**	= "0"	(Empty FIFO Access)
• **SIR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **SIR.RFCL**	= "1"	(Receive FIFO Critical Level)
• **FSR.RFNE**	= "1"	(Receive FIFO Not Empty)
• **FSR.RFCL**	= "1"	(Receive FIFO Critical Level)
• **FSR.RFO**	= "0"	(Receive FIFO Overrun)
• **FSR.RFFL[7:0]**	= "00000101"	(Receive FIFO Fill Level)

Message Buffer 0 / FIFO Buffer 1	Frame ID 2, channel A, from cycle 8
Message Buffer 1 / FIFO Buffer 2	Frame ID 5, channel A, from cycle 8
Message Buffer 2 / FIFO Buffer 3	Frame ID 7, channel A, from cycle 8
Message Buffer 3 / FIFO Buffer 4	Frame ID 2, channel A, from cycle 9
Message Buffer 4 / FIFO Buffer 5	Frame ID 5, channel A, from cycle 9
Message Buffer 5 / FIFO Buffer 3	empty
Message Buffer 6 / FIFO Buffer 4	empty
Message Buffer 7 / FIFO Buffer 5	empty

Figure 9: FIFO message buffer at the end of cycle 9, example two

4.5.2.5 Access to the FIFO during cycle 9

In cycle 9, after slot 9, the Host reads out the FIFO as described in chapter 4.4.3. Depending from the configuration of **OBCM.RDSS** and **OBCM.RHSS**, data and/or header section are transferred from the Message RAM to the Output Buffer.

If the Host triggers the CC to transfer the data section of the FIFO buffer from the Message RAM to the Output Buffer (**OBCM.RDSS** = "1"), always the **configured** payload length of the FIFO buffer is transferred (**WRHS2.PLC[6:0]**, **RDHS2.PLC[6:0]**, respectively), and not the payload length received and stored into the message buffer (**RDHS2.PLR[6:0]**). If the payload length received is smaller than the payload length configured, the remaining data bytes of the data section are filled with undefined data.

To access the FIFO, always the first FIFO buffer, referenced by **MRC.FFB[7:0]**, must be read out. In this example, the first FIFO buffer which must be always read out by the Host is buffer 0 (**MRC.FFB** = 0)..

```
/* write32bit(address, value) */
write32bit(0x0710, 0x00000003);   /* Write Output Buffer Command Mask
                                     OBCM.RHSS, OBCM.RDSS */

while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000200);   /* Request transfer of first FIFO message
                                     buffer MRC.FFB[7:0] to OBF Shadow by
                                     writing OBCR.OBRS[6:0] and OBCR.REQ */
```

Note: After the "Request Message RAM Transfer" command (**OBCR.REQ** = "1", **OBCR.OBRS** = "11"), the following FIFO related status flags change (see table 1):

- **SIR.RFCL** = "0" (Receive FIFO Critical Level)
- **FSR.RFCL** = "0" (Receive FIFO Critical Level)
- **FSR.RFFL[7:0]** = "00000100" (Receive FIFO Fill Level)

```
while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000300);   /* Toggle OBF Shadow and OBF Host by
                                     writing OBCR.VIEW = "1"
                                     Request transfer of first FIFO message
                                     buffer MRC.FFB[7:0] to OBF Shadow by
                                     writing OBCR.OBRS[6:0] and OBCR.REQ */


value32bit = read32bit(0x0700);   /* Read out transferred message buffer
                                     by reading RDHS1 (and RDHS2..3,
                                     RDDSn, MBS). The RDHS1 register
                                     only holds the frame
                                     ID of the received frame
                                     (RDHS1.FID[10:0]), all other bits
                                     are read as "0" */
```

After each of these transfer requests, FIFO related status flags change as described in the following table (**n** is the point in time when the Host triggers the CC for the first transfer from the Message RAM to the Output Buffer).

	Request Message RAM Transfer	**FIFO related status flags after the "Request Message RAM Transfer" command**								
Time	**OBCR**	**EIR. RFO**	**EIR. EFA**	**SIR. RFNE**	**SIR. RFCL**	**FSR. RFNE**	**FSR. RFCL**	**FSR. RFO**	**FSR. RFFL**	**RDHS1. FID**
n-1	-	0	0	1	1	1	1	0	5	-
n	`0x00000200`	0	0	1	0	1	0	0	4	-
n+1	`0x00000300`	0	0	1	0	1	0	0	3	2 (cycle 8)
n+2	`0x00000300`	0	0	1	0	1	0	0	2	5 (cycle 8)
n+3	`0x00000300`	0	0	1	0	1	0	0	1	7 (cycle 8)
n+4	`0x00000300`	0	0	0	0	0	0	0	0	2 (cycle 9)
n+5	`0x00000100`	0	0	0	0	0	0	0	0	5 (cycle 9)

Table 2: FIFO status flags during read out, example two

4.5.2.6 Empty FIFO Access

After the fifth transfer Message RAM to Output Buffer request by the Host, the FIFO is empty (**FSR.RFFL[7:0]** = "00000000", see table 2.) If after that point in time the Host triggers another transfer from the Message RAM to the Output Buffer from the FIFO, the flag "Empty FIFO Access" is set (**EIR.EFA** = "1").

```
value32bit = read32bit(0x0318);
```

Note: The value of the Receive FIFO Fill Level is **FSR.RFFL[7:0]** = "00000000"

```
write32bit(0x0710, 0x00000003);    /* Write Output Buffer Command Mask
                                      OBCM.RHSS, OBCM.RDSS */

while((read32bit(0x0714) & 0x00008000) != 0);    /* Wait until OBCR.OBSYS
                                                    is reset */

write32bit(0x0714, 0x00000200);    /* Request transfer of first FIFO message
                                      buffer MRC.FFB[7:0] to OBF Shadow by
                                      writing OBCR.OBRS[6:0] and OBCR.REQ */
```

Note: After the Host has triggered the transfer from the Message RAM to the Output Buffer from the empty FIFO, the flag **EIR.EFA** is set to "1".

5. List of Tables

FIFO status flags during read out, example one 23

FIFO status flags during read out, example two 32

APP_LOT.fm

6. List of Figures

Figure 1: FIFO status: empty, not empty, overrun. 7

Figure 2: FlexRay channel traffic example one 15

Figure 3: Message RAM structure for application example one 16

Figure 4: Content of FIFO message buffers at the end of cycle 8, example one . . . 19

Figure 5: Content of FIFO message buffers at the end of cycle 9, example one . . . 21

Figure 6: FlexRay channel traffic example two 24

Figure 7: Message RAM structure for application example two 25

Figure 8: Content of FIFO message buffers at the end of cycle 8, example two . . . 28

Figure 9: FIFO message buffer at the end of cycle 9, example two. 30