E-Ray

FlexRay IP Module

Application Note AN003

Message RAM Configuration

Date: December 3rd, 2007

for IP Revision 1.0.2

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.

APP_cover.fm

1. About this Document

1.1 Change Control

1.1.1 Current Status

Version 1.0.4

1.1.2 Change History

Issue	Date	By	Change
Version 1.0.0	03.05.2006	Kay Hammer	Initial Draft for IP Revision 1.0 RC 1
Version 1.0.1	19.05.2006	Kay Hammer	Update for IP Revision 1.0
Version 1.0.2	03.11.2006	Kay Hammer	Update for E-Ray Specification 1.2.3
Version 1.0.3	01.06.2007	Kay Hammer	Update for E-Ray Specification 1.2.5
Version 1.0.4	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

2. Introduction

This application note describes the configuration of the E-Ray Message RAM as part of the configuration procedure of an E-Ray FlexRay CC in POC state CONFIG.

The methods of reconfiguration message buffers during runtime are not part of this application note.

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. Message RAM Configuration

4.1 Overview

The Message RAM is cleared (all bits are written to "0") after an external reset or by CHI command CLEAR_RAMS (**SUCC1.CMD[3:0]** = 1100). During initialization of the Message RAM the flag **MHDS.CRAM** is set to "1" by the CC. The Host has to wait until the CC has reset **MHDS.CRAM** to "0" before the configuration of the message buffers in the Message RAM is started.

After an external reset, the CC is in POC state DEFAULT_CONFIG (**CCSV.POCS[5:0]** = 00 0000). It is recommend to enter POC state CONFIG with CHI command **SUCC1.CMD[3:0]** = 0001 before the configuration of the CC.

The application specific partitioning of the Message RAM, including the number of message buffers, assignment to static segment, dynamic segment, and FIFO, as well as the partitioning of the data section have to be fixed by the application programmer.

4.2 Message RAM Configuration Register (MRC)

The Message RAM Configuration Register defines the number of message buffers assigned to the static segment, dynamic segment and FIFO. The register can be written during DEFAULT_CONFIG or CONFIG state only.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MRC	R	0	0	0	0	0	SPLM*	SEC1*	SEC0*	LCB7*	LCB6*	LCB5*	LCB4*	LCB3*	LCB2*	LCB1*	LCB0*
0x0300	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	FFB7*	FFB6*	FFB5*	FFB4*	FFB3*	FFB2*	FFB1*	FFB0*	FDB7*	FDB6*	FDB5*	FDB4*	FDB3*	FDB2*	FDB1*	FDB0*
	W																
Reset		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

FDB[7:0] First Dynamic Buffer

0 = No group of buffers exclusively for the static segment configured

≥128 = No dynamic buffers configured

It is recommend to configure only message buffers belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, see chapter 4.2.1.4 for details.

FFB[7:0] First Buffer of FIFO

0 = All message buffers assigned to the FIFO

≥128 = No message buffer assigned to the FIFO

LCB[7:0] Last Configured Buffer

0…127= Number of message buffers is **LCB** + 1

≥128 = No message buffer configured

SEC[1:0] Secure Buffers

Not evaluated when the CC is in DEFAULT_CONFIG or CONFIG state. May be temporary unlocked, see chapter 4.2.1.6.

00 = Reconfiguration of message buffers enabled with numbers < **FFB**

Note: In case the node is configured as sync node (**SUCC1.TXSY** = "1"), message buffer 0 (in case **MRC.SPLM** = "1" message buffers 0 and 1) is reserved for sync frames and can be reconfigured only when the CC is in DEFAULT_CONFIG or CONFIG state.

01 = Reconfiguration of message buffers with numbers < **FDB** and with numbers ≥ **FFB** locked and transmission of message buffers for static segment with numbers ≥ **FDB** disabled

10 = Reconfiguration of all message buffers locked

11 = Reconfiguration of all message buffers locked
and transmission of message buffers for static segment with numbers ≥ **FDB** disabled

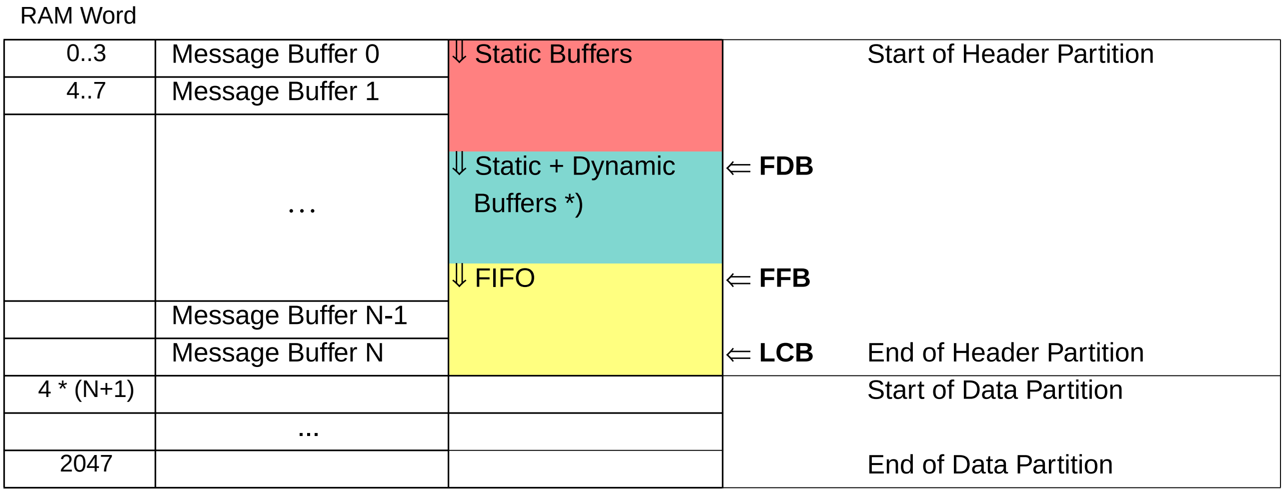
SPLM Sync Frame Payload Multiplex

This bit is evaluated only if the node is configured as sync node (**SUCC1.TXSY** = "1") or for single slot operation mode (**SUCC1.TSM** = "1"). If **MRC.SPLM** is set to "1" message buffers 0 and 1 are dedicated for sync or single slot frame transmission with different payload data on channel A and B. When this bit is set to "0", sync and single slot frames are transmitted from message buffer 0 with the same payload on both channels. Note that the channel filter configuration for message buffer 0 resp. message buffer 1 has to be chosen accordingly.

1 = Both message buffers 0 and 1 are locked against reconfiguration

0 = Only message buffer 0 locked against reconfiguration

Note: In case the node is configured as sync node (**SUCC1.TXSY** = "1") or for single slot mode operation (**SUCC1.TSM** = "1"), message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID



RAM Word				
0..3	Message Buffer 0	⇓ Static Buffers		Start of Header Partition
4..7	Message Buffer 1			
	…	⇓ Static + Dynamic Buffers *)	⇐ **FDB**	
		⇓ FIFO	⇐ **FFB**	
	Message Buffer N-1			
	Message Buffer N		⇐ **LCB**	End of Header Partition
4 * (N+1)				Start of Data Partition
	...			
2047				End of Data Partition

Figure 1: Message RAM structure

*) It is recommend to configure only message buffers belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, see chapter 4.2.1.4 for details.

4.2.1 Configuration Hints

4.2.1.1 General

To configure message buffers which are not exclusively reserved for static buffers, i.e. the header partition holds header sections of dynamic buffers, **MRC.FDB** has to be configured.

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.

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!**

4.2.1.2 Key Slot ID

The frame ID which is configured within the header section of message buffer 0 is the key slot ID.

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.

The frame configured with the key slot ID can be configured as startup frame, as sync frame, as single slot frame or as normal frame.

In case the frame configured with the key slot ID ought to be a startup frame, the flags **SUCC1.TXSY** and **SUCC1.TXST** must be configured to "1".

In case the frame configured with the key slot ID ought to be a sync frame, the flag **SUCC1.TXSY** must be configured to "1" and the flag **SUCC1.TXST** must be configured to "0".

In case the frame configured with the key slot ID ought to be a single slot frame (the CC only transmit in the pre-configured key slot), the flag **SUCC1.TSM** must be configured to "1". A single slot frame may be a sync or startup frame, too. **SUCC1.TXST** and **SUCC1.TXSY** have to be programmed accordingly. In ALL slots mode the CC may transmit in all slots. **SUCC1.TSM** is a configuration bit which can only be set/reset by the Host. The bit can be written in DEFAULT_CONFIG or CONFIG state only.

In case the frame configured with the key slot ID ought to be a normal frame, the flags **SUCC1.TXSY**, **SUCC1.TXST** and **SUCC1.TSM** must be configured to "0".

The configuration with **SUCC1.TXSY** = "0" and **SUCC1.TXST** = "1" is invalid.

Note: In case the node is configured as sync node (**SUCC1.TXSY** = "1") or for single slot mode operation (**SUCC1.TSM** = "1"), the message buffer 0 resp. 1 is reserved for sync frames or single slot frames and have to be configured with the node-specific key slot ID. In case the node is configured as sync node (**SUCC1.TXSY** = "1") or for single slot mode operation (**SUCC1.TSM** = "1"), message buffer 0 respectively message buffer 0,1 can be (re)configured in DEFAULT_CONFIG or CONFIG state only.

4.2.1.3 Different Payload for Startup, Sync and Single Slot Frames

In case a startup frame, sync frame or single slot frame shall contain different payload on channel A and channel B, two message buffers have to be configured.

The first message buffer must be message buffer 0. It has to be configured as transmit buffer with the key slot ID as frame ID, channel filter control for channel A only and the dedicated payload for channel A.

The second message buffer must be message buffer 1. It has to be configured as transmit buffer with the key slot ID as frame ID, channel filter control for channel B only and the dedicated payload for channel B.

In addition,
for a startup frame, **SUCC1.TXST** and **SUCC1.TXSY** have to be programmed to "1",
for a sync frame, **SUCC1.TXSY** has to be programmed to "1",
for a single slot frame, **SUCC1.TSM** has to be programmed to "1". A single slot frame may be a sync or startup frame, too. **SUCC1.TXST** and **SUCC1.TXSY** have to be programmed accordingly.

It is recommend to set **MRC.SPLM** = "1" to lock both message buffers 0 and 1 against reconfiguration.

4.2.1.4 Cycle Counter Filtering

It is recommend to configure only message buffers with message IDs belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, **MRC.LCB** resp.. Otherwise, the search algorithm of the message handler can not work in its most efficiently way.

In case two or more message buffers are assigned to the same slot by use of cylce filtering, all of them must be located in the header partition area reserved for static segments only (all message buffers < **MRC.FDB**, < **MRC.FFB** resp.) if the message ID belongs to the static segment, or all of them must be located in the header partition area reserved for (static and) dynamic frames if the message ID belongs to the dynamic segment.

Note: It must be prohibited that message buffers with the same static frame ID, which are configured for cycle counter filtering, are located in the header partition area reserved for static segments only **and** in the header partition area reserved for (static and) dynamic frames at the same time.

4.2.1.5 Header and Data Section

The maximum number of header sections is 128. This means a maximum of 128 message buffers can be configured. The maximum length of a data section is 254 bytes. The length of the data section may be configured differently for each message buffer.
The restriction for the Message RAM configuration is that header partition and data partition may not

occupy more than 2048 33-bit words. Some maximum values are:

- 30 message buffers with 254 byte data section each
- 56 message buffers with 128 byte data section each
- 128 message buffers with 48 byte data section each

Note: The payload length configured and the length of the data section need to be configured identical for all message buffers belonging to the FIFO via **WRHS2.PLC[6:0]** and **WRHS3.DP[10:0]**. When the CC is not in POC state DEFAULT_CONFIG or CONFIG, reconfiguration of message buffers belonging to the FIFO is looked.
The look can be circumvented, for example if a parity error in the header section of a locked message buffer makes a transfer from the Input Buffer to the locked buffer Header Section neccessary.

4.2.1.6 Temporary unlocking of Header Section

For a transfer from the Input Buffer to the locked buffer Header Section 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, 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[10:0]**, and will be updated with new data.

4.2.2 Header Section Configuration Example A

The number of messages buffers in this example is 16. Therefore, **MRC.LCB** = 15.

The header partition contains only header sections of static message buffers (inclusive key slot transmission buffer). Therefore, dynamic and FIFO buffers are disabled by programming **MRC.FDB** = 128 and **MRC.FFB** = 128.

```
/***************************************************************************
* Source code section - write MRC register                                 *
***************************************************************************/
/* write32bit(address, value); */

write32bit(0x0300, 0x000F8080);  /* SPLM = 0, SEC[1:0] = 0, LCB = 15,
                                    FFB = 128, FDB = 128 */
```

The Message RAM is configured as follows:

RAM Word				
0..3	Message Buffer 0	⇓ Static Buffers		Start of Header Partition
4..7	Message Buffer 1			
	…			
56..59	Message Buffer 14			
60..63	Message Buffer 15		⇐ **LCB**	End of Header Partition
64				Start of Data Partition
	...			
2047				End of Data Partition

Figure 2: Message RAM structure from header section configuration example A

4.2.3 Header Section Configuration Example B

The number of messages buffers in this example is 4. Therefore, **MRC.LCB** = 3.

The header partition contains 2 static message buffers (including key slot transmission buffer), no dynamic message buffers and 2 FIFO buffers. The dynamic message buffers are disabled by programming **MRC.FDB** = 128. The FIFO size is configured to 2 message buffers by programming **MRC.FFB** = 2 and **MRC.LCB** = 3.

```
/***************************************************************************
* Source code section - write MRC register                                *
***************************************************************************/
/* write32bit(address, value); */

write32bit(0x0300, 0x00030280);  /* SPLM = 0, SEC[1:0] = 0, LCB = 3,
                                    FFB = 2, FDB = 128 */
```

The Message RAM is configured as follows:

RAM Word				
0..3	Message Buffer 0	⇓ Static Buffers		Start of Header Partition
4..7	Message Buffer 1			
8..11	Message Buffer 2	⇓ FIFO	⇐ **FFB**	
12..15	Message Buffer 3		⇐ **LCB**	End of Header Partition
16				Start of Data Partition
	...			
2047				End of Data Partition

Figure 3: Message RAM structure from header section configuration example B

4.2.4 Header Section Configuration Example C

The number of messages buffers in this example is 10. Therefore, **MRC.LCB** = 9.

The header partition contains 1 static message buffer (key slot transmission buffer), 4 dynamic message buffers and 5 FIFO buffers. The start of the header section of dynamic message buffers is configured by **MRC.FDB** = 1. Since **MRC.FFB** = 5, the number of dynamic message buffers is 4. The FIFO size is configured to 5 message buffers by programming **MRC.FFB** = 5 and **MRC.LCB** = 9.

```
/***************************************************************************
* Source code section - write MRC register                               *
***************************************************************************/
/* write32bit(address, value); */

write32bit(0x0300, 0x00090501);  /* SPLM = 0, SEC[1:0] = 0, LCB = 9,
                                    FFB = 5, FDB = 1 */
```

The Message RAM is built up as follows:

RAM Word				
0..3	Message Buffer 0	Static Buffer		Start of Header Partition
4..7	Message Buffer 1	⇩ Static + Dynamic	⇐ **FDB**	
8..11	Message Buffer 2	Buffers *)		
12..15	Message Buffer 3			
16..19	Message Buffer 4			
20..23	Message Buffer 5	⇩ FIFO	⇐ **FFB**	
24..27	Message Buffer 6			
28..31	Message Buffer 7			
32..35	Message Buffer 8			
36..39	Message Buffer 9		⇐ **LCB**	End of Header Partition
40				Start of Data Partition
	...			
2047				End of Data Partition

Figure 4: Message RAM structure from header section configuration example C

*) It is recommend to configure only message buffers belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, see chapter 4.2.1.4 for details.

4.2.5 Header Section Configuration Example D

The number of messages buffers is this example is 128. Therefore, **MRC.LCB** = 127.

The header partition contains 1 static message buffer (key slot transmission buffer) and 127 FIFO buffers. The FIFO size is configured to 127 message buffers by programming **MRC.FFB** = 1 and **MRC.LCB** = 127.

```
/***************************************************************************
* Source code section - write MRC register                                *
***************************************************************************/
/* write32bit(address, value); */

write32bit(0x0300, 0x007F0180);  /* SPLM = 0, SEC[1:0] = 0, LCB = 127,
                                    FFB = 1, FDB = 128 */
```

The Message RAM is configured as follows

RAM Word				
0..3	Message Buffer 0	Static Buffer		Start of Header Partition
4..7	Message Buffer 1	⇓ FIFO	⇐ **FFB**	
...	...			
508..511	Message Buffer 127		⇐ **LCB**	End of Header Partition
512				Start of Data Partition
	...			
2047				End of Data Partition

Figure 5: Message RAM structure from header section configuration example D

4.3 Data Pointer Configuration

The data section of a message buffer is referenced by the data pointer configured in the header section. This data pointer has to be configured in **WRHS3.DP[10:0]** for each message buffer. The data pointer should not be modified during runtime. For message buffers belonging to the receive FIFO configuration is possible in DEFAULT_CONFIG or CONFIG state only.

The data partition starts after the last word of the header partition at word number (**MRC.LCB** + 1) * 4. Therefore, the data pointer of the message buffer must be greater than or equal to (**MRC.LCB** + 1) * 4. When configuring the message buffers in the Message RAM the programmer has to assure that the data pointers point to addresses within the data partition.

The beginning and the end of a message buffers data section is determined by the data pointer and the payload length configured in the message buffers header section, respectively. This enables a flexible usage of the available RAM space for storage of message buffers with different data length. It is recommend but not essential to set the datapointers back-to-back and consecutively.

Different message buffers may have the same data pointer configured. Unintended overwriting of the data sections by a erroneous configuration of the data pointers from different message buffers is not prevented.

The header section of the Message RAM is secured against overwritting. Therefore, if the data pointer of a message buffer is configured to point in the header section, or if the remaining RAM space between data pointer and the end of the Message RAM is smaller than the configured payload of the respective message buffer, the respective parts of the payload, which overlaps the secured header section, will not be stored and gets lost. In this cases, the error flag **MHDF.WAHP** is set.

The application programmer has to assure that data sections from different message buffers are not overlapping. The data pointer can only point to the start of a 32-bit word in the Message RAM. Therefore, if the size of the configured payload length is an odd number of 2-byte words, the remaining 2-byte word in the Message RAM is unused and inaccesible.

In case of an odd payload length (PLC = 1,3,5,...) the application has to write zero to the last 16 bit of the message buffer data section to ensure that the padding pattern is all zero.

4.3.1 Data Pointer Configuration Example A

The number of messages buffers in this example is 3. Therefore, **MRC.LCB** = 2. All data pointers must be greater than or equal to (**MRC.LCB** + 1) * 4 = 12.

In the header partition only static message buffers are configured. Dynamic and FIFO message buffers are disabled by programming **MRC.FDB** = 128 and **MRC.FFB** = 128.

The configured static frame data length is **MHDC.SFDL[6:0]** = 3 (two byte words).

The configured data pointer for message buffer 0 is **MB0.WRHS3.DP[10:0]** = 12.
The configured data pointer for message buffer 1 is **MB1.WRHS3.DP[10:0]** = 14.
The configured data pointer for message buffer 2 is **MB2.WRHS3.DP[10:0]** = 16.

The Message RAM is configured as follows:

RAM Word					
0..3	Message Buffer 0	Static Buffer, DP = 12			Start of Header Partition
4..7	Message Buffer 1	Static Buffer, DP = 14			
8..11	Message Buffer 2	Static Buffer, DP = 16		⇐ **LCB**	End of Header Partition
12		MB0 Data1	MB0 Data0	⇐ **DP MB0**	Start of Data Partition
13		unused *)	MB0 Data2		
14		MB1 Data1	MB1 Data0	⇐ **DP MB1**	
15		unused *)	MB1 Data2		
16		MB2 Data1	MB2 Data0	⇐ **DP MB2**	
17		unused *)	MB2 Data2		
18..2046	...	unused			
2047		unused			End of Data Partition

Figure 6: Message RAM structure from data pointer configuration example A

*) inaccessible

4.3.2 Data Pointer Configuration Example B

The number of messages buffers in this example is 4. Therefore, **MRC.LCB** = 3. All data pointers must be greater than or equal to (**MRC.LCB** + 1) * 4 = 16.

The header partition contains 2 static message buffers, 2 dynamic message buffers and no FIFO buffer. The start of the header section of dynamic message buffers is configured by **MRC.FDB** = 2. Since no FIFO buffers are configured and **MRC.LCB** = 3, the number of dynamic message buffers is 2. The FIFO buffers are disabled by programming **MRC.FFB** = 128.

The configured static frame data length is **MHDC.SFDL[6:0]** = 3 (two-byte words).
The configured payload length for the dynamic message buffers is **MB2,3.WRHS2.PLC[6..0]** = 4 two-byte words.

The configured data pointer for message buffer 0 is **MB0.WRHS3.DP[10:0]** = 2046.
The configured data pointer for message buffer 1 is **MB1.WRHS3.DP[10:0]** = 2044.
The configured data pointer for message buffer 2 is **MB2.WRHS3.DP[10:0]** = 2042.
The configured data pointer for message buffer 3 is **MB3.WRHS3.DP[10:0]** = 2040.

The Message RAM is configured as follows

RAM Word					
0..3	Message Buffer 0	Static Buffer, DP = 2046			Start of Header Partition
4..7	Message Buffer 1	Static Buffer, DP = 2044			
8..11	Message Buffer 2	Dyn. Buffer, DP = 2042		⇐ **FDB**	
12..15	Message Buffer 3	Dyn. Buffer, DP = 2040		⇐ **LCB**	End of Header Partition
16..2039	...	unused			Start of Data Partition
2040		MB3 Data1	MB3 Data0	⇐ **DP MB3**	
2041		MB3 Data3	MB3 Data2		
2042		MB2 Data1	MB2 Data0	⇐ **DP MB2**	
2043		MB2 Data3	MB2 Data2		
2044		MB1 Data1	MB1 Data0	⇐ **DP MB1**	
2045		unused *)	MB1 Data2		End of Data Partition
2046		MB0 Data1	MB0 Data0	⇐ **DP MB0**	
2047		unused *)	MB0 Data2		End of Data Partition

Figure 7: Message RAM structure from data pointer configuration example B

*) inaccessible
It is recommend to configure only message buffers belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, see chapter 4.2.1.4 for details.

4.3.3 Data Pointer Configuration Example C

The number of messages 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 2 static message buffers, 2 dynamic message buffers and 4 FIFO buffers. The start of the header section of dynamic message buffers is configured by **MRC.FDB** = 2. Since **MRC.FFB** = 4, the number of dynamic message buffers is 2. The FIFO size is configured to 4 message buffers by programming **MRC.FFB** = 4 and **MRC.LCB** = 7.

The configured static frame data length is **MHDC.SFDL[6:0]** = 2 (two-byte words).
The configured payload length for the dynamic message buffers is **MB2,3.WRHS2.PLC[6:0]** = 3 (two-byte words).
The configured payload length for the FIFO buffers is **MB4,5,6,7.WRHS2.PLC[6:0]** = 4 (two-byte words).

The configured data pointer for message buffer 0 is **MB0.WRHS3.DP[10:0]** = 1100.
The configured data pointer for message buffer 1 is **MB1.WRHS3.DP[10:0]** = 1101.
The configured data pointer for message buffer 2 is **MB2.WRHS3.DP[10:0]** = 1985.
The configured data pointer for message buffer 3 is **MB3.WRHS3.DP[10:0]** = 1987.
The configured data pointer for message buffer 4 is **MB4.WRHS3.DP[10:0]** = 79.
The configured data pointer for message buffer 5 is **MB4.WRHS3.DP[10:0]** = 81.
The configured data pointer for message buffer 6 is **MB4.WRHS3.DP[10:0]** = 83.
The configured data pointer for message buffer 7 is **MB4.WRHS3.DP[10:0]** = 85.

The Message RAM is configured as follows:

RAM Word					
0..3	Message Buffer 0	Static Buffer, DP = 1100			Start of Header Partition
4..7	Message Buffer 1	Static Buffer, DP = 1101			
8..11	Message Buffer 2	Dyn. Buffer, DP = 1985		⇐ **FDB**	
12..15	Message Buffer 3	Dyn. Buffer, DP = 1987			
16..19	Message Buffer 4	FIFO, DP = 79		⇐ **FFB**	
20..23	Message Buffer 5	FIFO, DP = 81			
24..27	Message Buffer 6	FIFO, DP = 83			
28..31	Message Buffer 7	FIFO, DP = 85		⇐ **LCB**	End of Header Partition
32		unused			Start of Data Partition
33..78	...	unused			
79		MB4 Data1	MB4 Data0	⇐ **DP MB4**	
80		MB4 Data3	MB4 Data2		
81		MB5 Data1	MB5 Data0	⇐ **DP MB5**	
82		MB5 Data3	MB5 Data2		
83		MB6 Data1	MB6 Data0	⇐ **DP MB6**	
84		MB6 Data3	MB6 Data2		
85		MB7 Data1	MB7 Data0	⇐ **DP MB7**	
86		MB7 Data3	MB7 Data2		
87..1099	...	unused			
1100		MB0 Data1	MB0 Data0	⇐ **DP MB0**	
1101		MB1 Data1	MB1 Data0	⇐ **DP MB1**	
1102..1984	...	unused			
1985		MB2 Data1	MB2 Data0	⇐ **DP MB2**	
1986		unused *)	MB2 Data2		
1987		MB3 Data1	MB3 Data0	⇐ **DP MB3**	
1988		unused *)	MB3 Data2		
1989..2046	...	unused			
2047		unused			End of Data Partition

Figure 8: Message RAM structure from data pointer configuration example C

*) inaccessible

It is recommend to configure only message buffers belonging to the dynamic segment in the header partition area defined by **MRC.FDB** and **MRC.FFB**, see chapter 4.2.1.4 for details.

APP_description.fm

5. List of Figures

APP_LOF.fm