E-Ray

FlexRay IP Module

Application Note AN005

Handling of Parity Errors

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

APP_TOC.fm

1. About this Document

1.1 Change Control

1.1.1 Current Status

Version 1.0.2

1.1.2 Change History

Issue	Date	By	Change
Version 1.0.0	31.01.2007	Kay Hammer	Initial Draft for IP Revision 1.0.1
Version 1.0.1	01.06.2007	Kay Hammer	Update for E-Ray Specification 1.2.5
Version 1.0.2	03.12.2007	Kay Hammer	Update for E-Ray 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
3	Robert Bosch GmbH	E-Ray Module Integration Guide Rev. 1.0.3

1.3 Terms and Abbreviations

This document uses the following terms and abbreviations:

Term	Meaning
CC	Communication Controller

APP_about.fm

2. Introduction

This application note describes the parity checking mechanism implemented in the E-Ray core and shows possible ways of handling announced parity errors.

3. Parity Checker Procedure

3.1 Principle of Parity Checks

There is a parity checking mechanism implemented in the E-Ray core to assure the integrity of the data stored in the RAM blocks. The E-Ray core uses an even parity. That means that if the number of counted digit "1" which appears in the respective data word is even, a "0" parity bit is generated and stored together with the data word (32 + 1 bit). If the number of counted digit "1" which appear in the respective data word is odd, a "1" parity bit is generated and stored together with the data word (32 + 1 bit). Therefore, the number of digit "1" in the 33 bit width RAM word is always even.

3.2 E-Ray Parity Checker

3.2.1 Implementation

The E-Ray parity checking mechanism implemented in the E-Ray core assures the integrity of the data stored in the seven RAM blocks. The RAM blocks have parity generator / checker attached as shown in figure 1. When data is written to a RAM block, the local parity generator generates the parity bit. The E-Ray core uses an even parity. The parity bit is stored together with the respective data word (see chapter 3.1). The parity is checked each time a data word is read from any of the RAM blocks. The E-Ray core’s internal data buses have a width of 32 bits.

3.2.2 RAM Organisation

The seven RAM blocks of the E-Ray core are:

- Input Buffer RAM Shadow
- Input Buffer RAM Host
- Output Buffer RAM Shadow
- Output Buffer RAM Host
- Transient Buffer RAM Channel A
- Transient Buffer RAM Channel B
- Message RAM

Figure 1 shows the data paths between the RAM blocks and the parity generators / checkers.

Note: The header section of a message is stored, transferred resp. separately from the data section of a message in all RAMs expect of the Message RAM. For the two Input Buffer RAMs, the two Output Buffer RAMs and the two Transient Buffer RAMs the header information of a message is stored via FFs.

Note: Parity generator and checker are not part of the RAM blocks. They are positioned between RAM and E-Ray core.

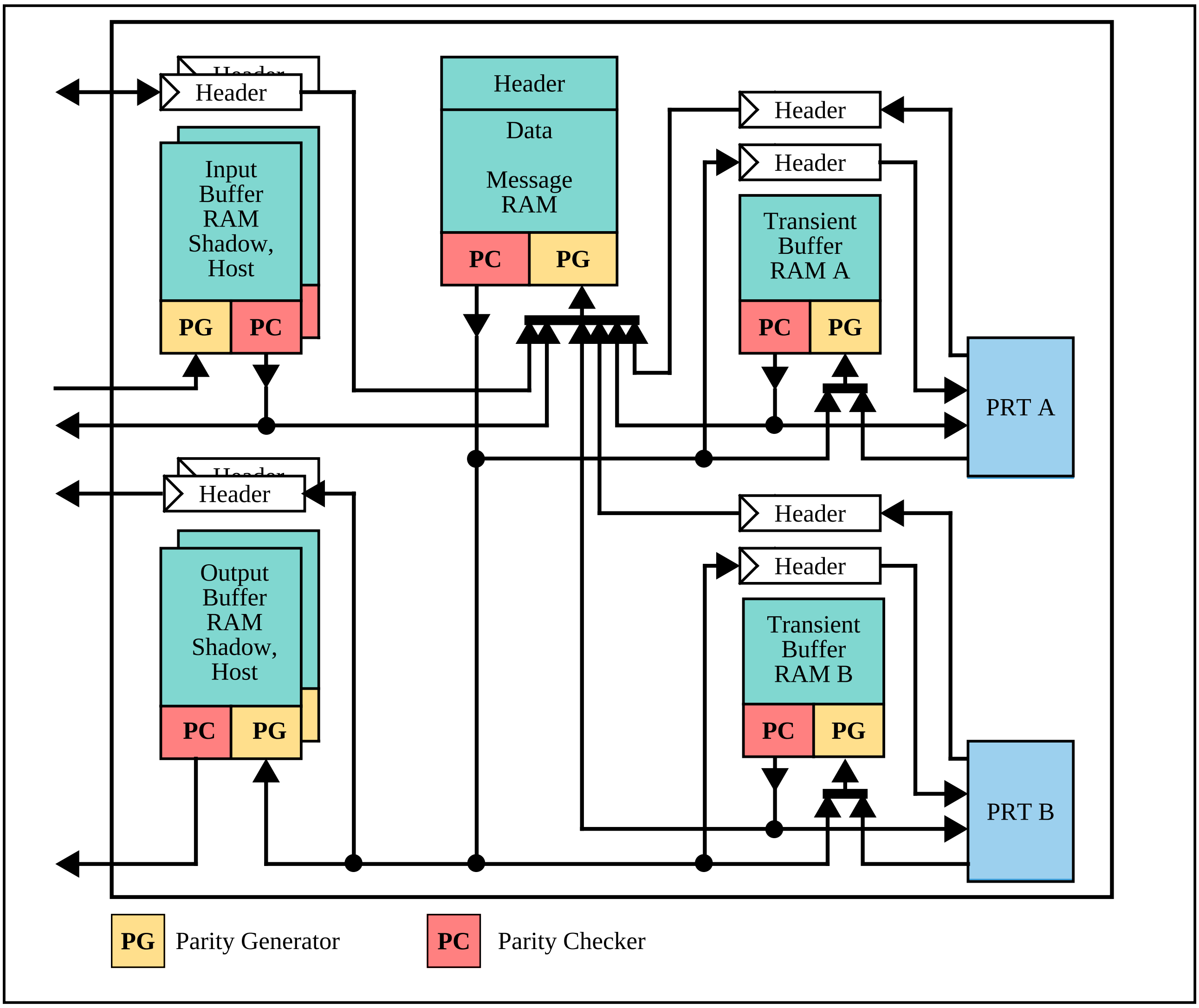

Figure 1: Parity generation and check

3.2.3 Message Handler Status Register (MHDS)

If a parity error is detected, the respective error flag is set. The parity error flags **MHDS.PIBF**, **MHDS.POBF**, **MHDS.PMR**, **MHDS.PTBF1**, **MHDS.PTBF2**, and the faulty message buffer indicators **MHDS.FMBD**, **MHDS.MFMB** and **MHDS.FMB[6:0]** are located in the Message Handler Status register. These single error flags control the error interrupt flag **EIR.PERR**.

Bit		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MHDS	R	0	MBU6	MBU5	MBU4	MBU3	MBU2	MBU1	MBU0	0	MBT6	MBT5	MBT4	MBT3	MBT2	MBT1	MBT0
0x0310	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	FMB6	FMB5	FMB4	FMB3	FMB2	FMB1	FMB0	CRAM	MFMB	FMBD	PTBF2	PTBF1	PMR	POBF	PIBF
	W																
Reset		0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

A flag is cleared by writing "1" to the corresponding bit position. Writing a "0" has no effect on the flag. The register will also be cleared by hard reset or by CHI command CLEAR_RAMS.

PIBF Parity Error Input Buffer RAM

1 = Parity error occurred when reading Input Buffer RAM
0 = No parity error

POBF Parity Error Output Buffer RAM

1 = Parity error occurred when reading Output Buffer RAM
0 = No parity error

PMR Parity Error Message RAM

1 = Parity error occurred when reading the Message RAM
0 = No parity error

PTBF1 Parity Error Transient Buffer A

1 = Parity error occurred when reading Transient Buffer RAM A
0 = No parity error

PTBF2 Parity Error Transient Buffer B

1 = Parity error occurred when reading Transient Buffer RAM B
0 = No parity error

Note: When one of the flags **PIBF**, **POPF**, **PMR**, **PTBF1**, **PTBF2** changes from "0" to "1" **EIR.PERR** is set to "1".

FMBD Faulty Message Buffer Detected

1 = Message buffer referenced by **FMB[6:0]** holds faulty data due to a parity error
0 = No parity error

MFMB Multiple Faulty Message Buffers Detected

1 = Another faulty message buffer was detected while flag **FMBD** is set
0 = No additional faulty message buffer

CRAM Clear all internal RAMs

Signals that execution of the CHI command CLEAR_RAMS is ongoing (all bits of all internal RAM blocks are written to "0"). The bit is set by hard reset or by CHI command CLEAR_RAMS.

1 = Execution of the CHI command CLEAR_RAMS ongoing
0 = No execution of the CHI command CLEAR_RAMS

FMB[6:0] Faulty Message Buffer

Parity error occurred when reading from the message buffer or when transferring data from Input Buffer or Transient Buffer 1,2 to the message buffer referenced by **FMB[6:0]**. Value only valid when one of the flags **PIBF**, **PMR**, **PTBF1**, **PTBF2**, and flag **FMBD** is set. Is not updated while flag **FMBD** is set.

MBT[6:0] Message Buffer Transmitted

Number of last successfully transmitted message buffer. If the message buffer is configured for single-shot mode, the respective **TXR** flag in the **TXRQ1/2/3/4** registers was reset.

MBU[6:0] Message Buffer Updated

Number of message buffer that was updated last by the CC. For this message buffer the respective **ND** and / or **MBC** flag in the **NDAT1/2/3/4** registers and the **MBSC1/2/3/4** registers are also set.

Note: **MBT[6:0]** and **MBU[6:0]** are reset when the CC leaves CONFIG state or enters STARTUP state.

3.2.4 Interrupt Registers

Parity errors are signalled to the Host by the parity error flag from the **Error Interrupt Register (EIR)**:

EIR.PERR Parity Error

The flag signals a parity error to the Host. It is set whenever one of the flags MHDS.PIBF, MHDS.POBF, MHDS.PMR, MHDS.PTBF1, MHDS.PTBF2 changes from "0" to "1".

1 = Parity error detected

0 = No parity error detected

With the **Error Interrupt Line Select** (**EILS**) and **Error Interrupt Enable Set / Reset** (**EIES**, **EIER**) registers, interrupt generated by a parity error can be assigned to one module interrupt line. See Ref. 2 for details.

EILS.PERRL Parity Error Interrupt Line

1 = Interrupt assigned to interrupt line eray_int1

0 = Interrupt assigned to interrupt line eray_int0

EIES.PERRE/EIER.PERRE Parity Error Interrupt Enable/Disable

The enable bits are set by writing to address 0x0030 (**EIES**) and reset by writing to address 0x0034 (**EIER**). Writing a "1" sets/resets the specific enable bit, writing a "0" has no effect. Reading from both addresses will return the same value.

1 = Interrupt enabled

0 = Interrupt disabled

3.3 Appearance and Detection of Parity Errors

The E-Ray parity checking mechanism implemented in the E-Ray core assures the integrity of the data stored in the seven RAM blocks. The parity bit is stored together with the respective data word (see chapter 3.1). The parity is checked each time a data word is read from any of the RAM blocks. The parity checker is not able to detect which bit is invalid. Furthermore, the parity checker can only detect parity errors, and can not repair them.

Whenever one of the following accesses to a RAM block occurs, the parity is checked:

1) Data transfer between the Input Buffer RAM and the Message RAM
2) Data transfer between the Input Buffer RAM and the Host databus
3) Scan of Message RAM header partition
4) Data transfer between Message RAM and Transient Buffer RAM
5) Data transfer between Transient Buffer RAM and Protocol Controller
6) Data transfer between Transient Buffer RAM and Message RAM
7) Data transfer between Message RAM and Output Buffer RAM
8) Data transfer between the Output Buffer RAM and the Host databus
9) Data transfer from Transient Buffer RAM in case of Message Handler reads network management information (**PPI** = "1").

Note: Only during read access to RAM blocks, a parity error is detectable.

As mentioned in chapter 3.2.2, the header section of a message is stored, transferred resp. separately from the data section of a message in all RAMs expect of the Message RAM.

For the two Input Buffer RAMs, the two Output Buffer RAMs and the two Transient Buffer RAMs the header information of a message is stored in FFs.

A parity error may occur in only in RAM blocks. The following parity errors can be detected by the parity checker of the RAM blocks:

RAM Block	Detectable Parity Error
Input Buffer RAM (Shadow, Host)	Parity error in data section
Output Buffer RAM (Shadow, Host)	Parity error in data section
Transient Buffer RAM A	Parity error in data section
Transient Buffer RAM B	Parity error in data section
Message RAM	Parity error in header section, parity error in data section

Table 1: Detectable Parity Errors

3.4 Parity Error Signalling

3.4.1 General Error Flags

When a parity error has been detected the following actions will be performed:

- The respective parity error flag in the register **MHDS** is set
- The parity error flag **EIR.PERR** is set and, if enabled, a module interrupt to the Host will be generated.

3.4.2 Specific Error Flags

1) Parity error during data transfer from Input Buffer RAM => Message RAM

a) Transfer of header and data section or data section only:

- **MHDS.PIBF** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- The data section of the respective message buffer is updated with invalid data
- Transmit buffer: Transmission request for the respective message buffer is not set

b) Transfer of data section only:

Parity error when reading header section of respective message buffer from Message RAM.

Note: Even if only the data section of a message buffer is transferred, the CC accesses the header section of the dedicated message buffer in the Message RAM to get information about data pointer, payload length etc.. That means that in this case the header information is read from the Message RAM, not from FFs.

- **MHDS.PMR** bit is set
- **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
- **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
- The data section of the respective message buffer is not updated
- Transmit buffer: Transmission request for the respective message buffer is not set

2) Parity error during Host reading Input Buffer RAM
   - **MHDS.PIBF** bit is set

3) Parity error during scan of header sections in Message RAM
   - **MHDS.PMR** is set
   - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
   - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
   - Message buffer is skipped

4) Parity error during data transfer from Message RAM => Transient Buffer RAM A, B
   - **MHDS.PMR** bit is set
   - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
   - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
   - Frame is not transmitted, frames already in transmission are invalidated by setting the frame CRC to zero

5) Parity error during data transfer from Transient Buffer RAM A, B => Protocol Controller A, B
   - **MHDS.PTBF1,2** bit is set
   - Frames already in transmission are invalidated by setting the frame CRC to zero

6) Parity error during data transfer from Transient Buffer RAM A, B => Message RAM

a) Parity error when reading header section of respective message buffer from Message RAM

   - **MHDS.PMR** bit is set
   - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
   - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer
   - The data section of the respective message buffer is not updated

b) Parity error when reading Transient Buffer RAM A, B

   - **MHDS.PTBF1,2** bit is set
   - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
   - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer

7) Parity error during data transfer from Message RAM => Output Buffer RAM
   - **MHDS.PMR** bit is set
   - **MHDS.FMBD** bit is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer
   - **MHDS.FMB[6:0]** indicates the number of the faulty message buffer

8) Parity error during Host reading Output Buffer RAM
   - **MHDS.POBF** bit is set

9) Parity error during data read of Transient Buffer RAM A, B

   When a parity error occurs when the Message Handler reads a frame with network management information (**PPI** = "1") from the Transient Buffer RAM A, B the corresponding network management vector registers **NMV1..3** are not updated from that frame.

3.5 Approach to detected Parity Errors

A parity error is caused by a wrecked RAM cell. It is possible that the logic cell is not permanent destroyed, but contains an invalid value in this moment (e.g. because of electromagnetic disturbance). If the cell is not permanent destroyed, the next write access leads to a valid value (the cell is self-healing).

Cells which have the capability for self-healing can be located in Input Buffer RAM (Shadow and Host), Outpur Buffer RAM (Shadow and Host), Data Section of Message RAM, Transient Buffer RAM A and Transient Buffer RAM B. This RAMs are overwritten with the next write access to the disturbed bit(s) caused by Host access or by FlexRay communication.

That means, that an error in the data section of the Message RAM, the Input and Output Buffer RAM and the Transient Buffer RAM A and B may be self-curing, because the data section is regularly updated.

The header section of the Message RAM is not regularly updated. The reconfiguration of a message buffer may be prohibited by the CC outside POC state DEFAULT CONFIG, POC state CONFIG resp.. For example, this may be the case for secured buffers, message buffer 0, 1, or for FIFO buffers. In that cases, the application should allow the Host to bring the CC to POC state CONFIG for reconfiguration the faulty secured message buffers. Afterwards, the node can be re-integrated to the FlexRay cluster.

If the cell is permanent destroyed, the application must detect the RAM block which is affected and has to accomplish an error correction routine.

If a parity error occurs, in all cases the transmission of the respective frame will be prevented!

3.5.1 Temporary Unlocking of Header Section

A parity error in the header section of a locked message buffer can also be fixed by a transfer from the Input Buffer to the locked Header Section. For this 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.

3.5.2 Classification of Detected Parity Errors

For every detected parity error the parity error flag **EIR.PERR** is set and, if enabled, a module interrupt of the Host will be generated. Additionally, parity error flags in the **MHDS** register are set.

3.5.2.1 Parity Error Flag MHDS.PMR is set

If the parity error flag **MHDS.PMR** is set, a parity error in the Message RAM is signalled. The error may occur during read of the header section or during read of the data section in the Message RAM. The header section is read by the message handler whenever information hold in the header section is needed by the message handler. That is the case when:

- Only data section of a message should be transferred from the Input Buffer RAM to the Message

RAM, because in this case the header section of the dedicated message buffer is read to find out the data pointer and the configured payload length of this message buffer. That means, if a parity error is detected through the parity error flag **MHDS.PMR** and the preceding action was a data-section-only transfer from the Input Buffer RAM to the Message RAM, **the parity error is located in the header partition** of the Message RAM. In this case, the data section of the respective message buffer in the Message RAM is not updated. The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a failure message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

- A message should be transferred from the Message RAM to the Transient Buffer RAM A or B, because in this case the header section and the data section of the dedicated message buffer is read to transmit the information from the header and the data section to the Transient Buffer RAM A, B. That means, if a parity error is detected through the parity error flag **MHDS.PMR** and the preceding action was a transmit request of the dedicated message, **the parity error is located in the header and/or in the data partition** of the Message RAM. In this case, the frame is not transmitted. Frames already in transmission are invalidated by setting the frame CRC to zero. The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a failure message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

- A message should be transferred from the Transient Buffer RAM A or B to the Message RAM, because in this case the header section of the dedicated message buffer is read to find out the data pointer and the configured payload length of this message buffer. That means, if a parity error is detected through the parity error flag **MHDS.PMR** and the preceding action was a message reception, **the parity error is located in the header partition** of the Message RAM. In this case, the data section of the respective message buffer in the Message RAM is not updated. The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a failure message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

- A message buffer should be transferred from the Message RAM to the Output Buffer RAM, because in this case the header section and the data section of the dedicated message buffer is scanned to transmit information from the header and the data section to the Output Buffer RAM. That means, if a parity error is detected through the parity error flag **MHDS.PMR** and the preceding action was a Message RAM Transfer request (see Ref. 2, chapter 4.11.7 Output Buffer Command Request, **OBCR.REQ**) for the dedicated message, **the parity error is located in the header and/or in the data partition** of the Message RAM. **In this case, the Output Buffer RAM is written with invalid data, and/or in the FFs which holds header information invalid data is stored.** The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a failure message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

3.5.2.2 Parity Error Flag MHDS.PIBF is set

If the parity error flag **MHDS.PIBF** is set, a parity error in the Input Buffer RAM is signalled. In the Input Buffer RAM, only the data section of a message buffer is latched. The Input Buffer RAM is read when:

- A message (data section) shall be transferred from the Input Buffer RAM to the Message RAM. That means, if an parity error is detected through the parity error flag **MHDS.PIBF** and the preceding action was an Input Buffer Request Host (see Ref. 2, chapter 4.10.6, Input Buffer Command Request, **IBCR.IBRH[6:0]**) with load of header and data section (see Ref. 2, chapter 4.10.5 Input Buffer Command Mask, **IBCM.LHSH** and **IBCM.LDSH**), or data section only (see Ref. 2, chapter 4.10.5 Input Buffer Command Mask, **IBCM.LDSH**), **the parity error is located in the**

Input Buffer RAM i.e. in the data section of the message. In this case, the designated data partition of the Message RAM is written with invalid data. If the message buffer is a transmit buffer, the transmission request for the respective message buffer is not set. The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a failure message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

- When the Host reads the Input Buffer RAM via the Host databus (Interface). That means, if a parity error is detected through the parity error flag **MHDS.PIBF** and the preceding action was a reading access to the write data section (see Ref. 2, chapter 4.10.1 Write Data Section [1..64], **WRDS[1..64]**), **the parity error is located in the Input Buffer RAM i.e. in the data section of the message.** The data section of the message is invalid. No further error flags are set.

3.5.2.3 Parity Error Flag MHDS.POBF is set

If the parity error flag **MHDS.POBF** is set, a parity error in the Output Buffer RAM is signalled. In the Output Buffer RAM only the data section of a message buffer is latched. The Output Buffer RAM is read when:

- The Host reads the Output Buffer RAM via the Host databus (Interface). That means, if a parity error is detected through the parity error flag **MHDS.POBF** the parity error is located in the Output Buffer RAM.

3.5.2.4 Parity Error MHDS.PTBF1(2) is set

If the parity error flag **MHDS.PTBF1(2)** is set, a parity error in the Transient Buffer RAM A(B) is signalled. In the Transient Buffer RAM A(B), only the data section of a message is latched. The Transient Buffer RAM A(B) is read when:

- Data is transferred from the Transient Buffer RAM A(B) to the Protocol Controller A(B). That means, if a parity error is signalled through the parity error flag **MHDS.PTBF1(2)** and the preceding action was a message transmit request, **the parity error is located in the Transient Buffer RAM A(B)**. The data section of the message is invalid. If the frame is already in transmission, it is invalidated by the CC through setting the CRC to zero.

- Data is transferred from the Transient Buffer RAM A(B) to the Message RAM. That means, if a parity error is signalled through the parity error flag **MHDS.PTBF1(2)** and the preceding action was a message reception, **the parity error is located in the Transient Buffer RAM A(B)**. The data section of the message is invalid. In this case, the designated data partition of the Message RAM is written with invalid data. The error flag **MHDS.FMBD** is set to indicate that **MHDS.FMB[6:0]** points to a faulty message buffer. **MHDS.FMB[6:0]** indicates the number of the faulty message buffer.

Note: When a parity error occurs when the Message Handler reads a frame with network management information (**PPI** = "1") from the Transient Buffer RAM A, B the corresponding network management vector registers **NMV1..3** are not updated from that time.

3.5.3 Behaviour of a FIFO Buffer in Case of a Parity Error

In case a parity error occurs in the header section of the Message RAM belonging to a FIFO message buffer, accurate access to the messages stored in the FIFO is not possible any more! Every message which should be stored into the faulty message buffer gets irrecoverable lost. If a parity error occurs in the header section of the Message RAM belonging to a FIFO message buffer, the FIFO needs to be reconfigured in all cases!

3.5.4 Error Correction Routines

If a parity error was detected, the Host may try to localise the source of the error. Through word-wise read, the 32-bit word which is affected by the parity error in the Message RAM, in Input Buffer RAM or the Output Buffer RAM can be localised.

A parity error in the Message RAM can be bypassed through reconfiguration the data pointer of the affected message buffer, or through reconfiguration of the message buffer itself if the parity error is located in the header section of the Message RAM.

The reconfiguration of a message buffer may be prohibited by the CC outside POC state DEFAULT CONFIG, POC state CONFIG resp.. For example, this may be the case for message buffer 0, 1, or for FIFO buffers. See Ref. 2, Application Note 003 (Message RAM Configuration) and Application Note 004 (FIFO Configuration) for details.

In thoses cases, the application should allow the Host to bring the CC to POC state CONFIG for reconfiguration the faulty and secured message buffers. Afterwards, the node can be re-integrated to the FlexRay cluster.

A parity error in the header section of a locked message buffer can also be fixed by a transfer from the Input Buffer to the locked Header Section. For this 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.

3.6 Application Hints

A parity error is a serious hardware error. Therefore it is recommend to configure **EIR.PERR**, **EILS.PERRL** and **EIES.PERRE**, so that an parity error is certainly signalled to the Host.

4. List of Tables

APP_LOT.fm

5. List of Figures