



ICs for Communications

Multichannel Network Interface Controller for HDLC/PPP
with 256 Channels, 28 T1 or 21 E1 Framers, M13 Multiplexer and
T3 Framer

MUNICH256FM

PEB 20256M E Version 1.1

PEF 20256M E Version 1.1

Preliminary Data Sheet 11.99

| PEB 20256M E PEF 20256M E | | |
|--|---------------------------------|--|
| Revision History: | | Current Version: 11.99 |
| Previous Version: | | |
| Page (in previous Version) | Page (in current Version) | Subjects (major changes since last revision) |
| | | |
| | | |

For questions on technology, delivery and prices please contact the Infineon Technologies Offices in Germany or the Infineon Technologies Companies and Representatives worldwide: see our webpage at <http://www.infineon.com>

ABM®, AOP®, ARCOFI®, ARCOFI®-BA, ARCOFI®-SP, DigiTape®, EPIC®-1, EPIC®-S, ELIC®, FALC®54, FALC®56, FALC®-E1, FALC®-LH, IDEC®, IOM®, IOM®-1, IOM®-2, IPAT®-2, ISAC®-P, ISAC®-S, ISAC®-S TE, ISAC®-P TE, ITAC®, IWE®, MUSAC®-A, OCTAT®-P, QUAT®-S, SICAT®, SICOFI®, SICOFI®-2, SICOFI®-4, SICOFI®-4µC, SLICOFI® are registered trademarks of Infineon Technologies AG.

ACE™, ASM™, ASP™, POTSWIRE™, QuadFALC™, SCOUT™ are trademarks of Infineon Technologies AG.

Edition 11.99

**Published by Infineon Technologies AG,
SC,
Balanstraße 73,
81541 München**

© Infineon Technologies AG 1999.
All Rights Reserved.

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies Office.

Infineon Technologies AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you – get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components¹ of the Infineon Technologies AG, may only be used in life-support devices or systems² with the express written approval of the Infineon Technologies AG.

- 1 A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.
- 2 Life support devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

Preface

The Multichannel Network Interface Controller for HDLC is a Multichannel Protocol Controller for a wide area of telecommunication and data communication applications.

Organization of this Document

This Preliminary Data Sheet is divided into ten chapters. It is organized as follows:

- Chapter 1, MUNICH256FM Overview
Gives a general description of the product, lists the key features, and presents some typical applications.
- Chapter 2, Pin Description
Lists pin locations with associated signals, categorizes signals according to function, and describes signals.
- Chapter 3, General Overview
This chapter provides short descriptions of all MUNICH256FM internal function blocks.
- Chapter 4, Functional Description
Gives a detailed description about all functions supported by the MUNICH256FM.
- Chapter 5, Interface Description
This chapter provides functional diagrams of all interfaces.
- Chapter 6, Channel Programming / Reprogramming Concept
This chapter provides a detailed description of the channel programming concept.
- Chapter 7, Reset and Initialization procedure
Gives examples for MUNICH256FM initialization procedure and operation.
- Chapter 8, Register Description
Gives a detailed description about all MUNICH256FM on-chip registers.
- Chapter 9, Electrical Characteristics
Gives a detailed description of all electrical DC and AC characteristics and provides timing diagrams and values for all interfaces.
- Chapter 10, Package Outline

| | | |
|-------|---|----|
| 1 | MUNICH256FM Overview | 15 |
| 1.1 | General Features | 15 |
| 1.1.1 | M12 Multiplexer and DS2 Framer | 16 |
| 1.1.2 | M23 Multiplexer and DS3 Framer | 16 |
| 1.1.3 | Frame Alignment T1 Features | 17 |
| 1.1.4 | Signaling Controller T1 Features | 17 |
| 1.1.5 | Frame Alignment E1 Features | 17 |
| 1.1.6 | Signaling Controller E1 Features | 18 |
| 1.1.7 | Bit Error Rate Tester | 18 |
| 1.2 | Logic Symbol | 19 |
| 1.3 | General System Integration | 20 |
| 2 | Pin Description | 21 |
| 2.1 | PCI Bus Interface | 22 |
| 2.2 | SPI Interface | 28 |
| 2.3 | Local Microprocessor Interface | 29 |
| 2.4 | Serial Interface | 33 |
| 2.5 | Test Interface | 38 |
| 2.6 | Power Supply, Reserved Pins and No-connect Pins | 39 |
| 3 | General Overview | 41 |
| 3.1 | Functional Overview | 41 |
| 3.2 | Block Diagram | 42 |
| 3.3 | Internal Interface | 42 |
| 3.4 | Block Description | 43 |
| 4 | Functional Description | 47 |
| 4.1 | Port Handler | 47 |
| 4.1.1 | Local Port Loop | 48 |
| 4.1.2 | Remote Line Loops | 48 |
| 4.1.3 | Test Breakout | 50 |
| 4.2 | Time slot Handler | 51 |
| 4.2.1 | Channelized Modes | 51 |
| 4.2.2 | Unchannelized Mode | 52 |
| 4.3 | Data Management Unit | 53 |
| 4.3.1 | Descriptor Concept | 53 |
| 4.3.2 | Receive Descriptor | 54 |
| 4.3.3 | Data Management Unit Receive | 58 |
| 4.3.4 | Transmit Descriptor | 60 |
| 4.3.5 | Data Management Unit Transmit | 63 |
| 4.3.6 | Byte Swapping | 65 |
| 4.3.7 | Transmission Bit/Byte Ordering | 66 |
| 4.4 | Buffer Management | 67 |
| 4.4.1 | Internal Receive Buffer | 67 |

| | | |
|---------|---|----|
| 4.4.2 | Internal Transmit Buffer | 68 |
| 4.5 | Protocol Description | 71 |
| 4.5.1 | HDLC Mode | 71 |
| 4.5.2 | Bit Synchronous PPP with HDLC Framing Structure | 72 |
| 4.5.3 | Octet Synchronous PPP | 72 |
| 4.5.4 | Transparent Mode | 73 |
| 4.6 | T1 Framer and FDL Function | 74 |
| 4.6.1 | 4-Frame Multiframe | 74 |
| 4.6.2 | ESF Mode | 75 |
| 4.6.2.1 | Multiframe Synchronization Procedure of the Receiver | 76 |
| 4.6.2.2 | CRC-6 Generation / Check according to ITU-T G.706 | 76 |
| 4.6.2.3 | Remote Alarm (Yellow Alarm) Generation / Detection | 77 |
| 4.6.2.4 | Facility Data Link | 77 |
| 4.6.3 | SF Mode | 80 |
| 4.6.3.1 | Synchronization Procedure of the Receiver | 80 |
| 4.6.3.2 | Remote Alarm (Yellow Alarm) Generation / Detection | 81 |
| 4.6.4 | Common Features for SF and ESF | 82 |
| 4.6.4.1 | AIS (Blue Alarm) Generation/Detection | 82 |
| 4.6.4.2 | Loss of Signal (Red Alarm) Detection | 82 |
| 4.6.4.3 | In-Band Loop Generation and Detection | 83 |
| 4.6.4.4 | Pulse Density Detection | 83 |
| 4.6.4.5 | Error Performance Monitoring | 83 |
| 4.6.4.6 | Pseudo-random Bit Sequence Generator and Monitor | 84 |
| 4.7 | E1 Framing and Signaling | 85 |
| 4.7.1 | Doubleframe Format | 86 |
| 4.7.1.1 | Synchronization Procedure of the Receiver | 86 |
| 4.7.1.2 | A-bit Access | 87 |
| 4.7.1.3 | Sa-bit Access | 87 |
| 4.7.2 | CRC-4 Multiframe | 88 |
| 4.7.2.1 | Synchronization Procedure of the Receiver | 89 |
| 4.7.2.2 | CRC-4 Performance Monitoring | 89 |
| 4.7.2.3 | A-Bit Access | 90 |
| 4.7.2.4 | Sa-bit Access | 92 |
| 4.7.2.5 | E-Bit Access | 93 |
| 4.7.3 | Common Features for E1 Doubleframe and CRC-4 Multiframe | 94 |
| 4.7.3.1 | Error Performance Monitoring and Alarm Handling | 94 |
| 4.7.3.2 | Loss of Signal Detection | 95 |
| 4.7.3.3 | In-Band Loop Generation and Detection | 96 |
| 4.7.3.4 | Pseudo-random Bit Sequence Generator and Monitor | 96 |
| 4.8 | Signaling Controller Protocol Modes | 98 |
| 4.8.1 | HDLC Mode | 98 |
| 4.8.2 | Transparent Mode | 99 |
| 4.8.3 | BOM Mode | 99 |

| | | |
|----------|--|-----|
| 4.8.4 | Sa-bit Access | 100 |
| 4.8.5 | Signalling Controller FIFO Operations | 101 |
| 4.9 | M12 Multiplexer/Demultiplexer and DS2 framer | 105 |
| 4.9.1 | M12 multiplex format | 105 |
| 4.9.1.1 | Synchronization Procedure | 106 |
| 4.9.1.2 | Multiplexer/Demultiplexer | 106 |
| 4.9.1.3 | Loopback Control | 106 |
| 4.9.1.4 | Alarm Indication Signal | 107 |
| 4.9.2 | ITU-T G.747 format | 108 |
| 4.9.2.1 | Synchronization Procedure | 108 |
| 4.9.2.2 | Multiplexer/Demultiplexer | 108 |
| 4.9.2.3 | Parity Bit | 109 |
| 4.9.2.4 | Remote Alarm Indication | 109 |
| 4.9.2.5 | Alarm Indication Signal | 109 |
| 4.10 | M23 multiplexer and DS3 framer | 110 |
| 4.10.1 | M23 multiplex format | 110 |
| 4.10.1.1 | Synchronization Procedure | 111 |
| 4.10.1.2 | Multiplexer/Demultiplexer | 111 |
| 4.10.1.3 | X-bit | 111 |
| 4.10.1.4 | Alarm Indication Signal, Idle Signal | 112 |
| 4.10.1.5 | Loss of Signal | 112 |
| 4.10.1.6 | Performance Monitor | 112 |
| 4.10.2 | C-bit parity format | 113 |
| 4.10.2.1 | Synchronization Procedure | 114 |
| 4.10.2.2 | Multiplexer/Demultiplexer | 114 |
| 4.10.2.3 | X-bit | 114 |
| 4.10.2.4 | Far End Alarm and Control Channel | 115 |
| 4.10.2.5 | Path Maintenance Data Link Channel | 115 |
| 4.10.2.6 | Loopback Control | 115 |
| 4.10.2.7 | Alarm Indication Signal, Idle Signal | 115 |
| 4.10.2.8 | Loss of Signal | 116 |
| 4.10.2.9 | Performance Monitor | 116 |
| 4.10.3 | Full Payload Rate Format | 117 |
| 4.11 | Test Unit | 118 |
| 4.12 | Mailbox | 120 |
| 4.13 | Interrupt Controller | 122 |
| 4.13.1 | Layer Two interrupts | 122 |
| 4.13.1.1 | General Interrupt Vector Structure | 124 |
| 4.13.1.2 | System Interrupts | 126 |
| 4.13.1.3 | Port Interrupts | 127 |
| 4.13.1.4 | Channel Interrupts | 129 |
| 4.13.1.5 | Command Interrupts | 134 |
| 4.13.2 | Layer One Interrupts | 136 |

| | | |
|----------|--|------------|
| 4.13.2.1 | General Interrupt Vector Structure | 137 |
| 4.13.2.2 | T1/E1 Framer Interrupts | 138 |
| 4.13.2.3 | Facility Data Link Interrupts | 140 |
| 4.13.2.4 | DS3, DS2 and Test Unit Interrupts | 142 |
| 4.13.2.5 | Mailbox Interrupts | 145 |
| 5 | Interface Description | 146 |
| 5.1 | PCI Interface | 146 |
| 5.1.1 | PCI Read Transaction | 146 |
| 5.1.2 | PCI Write Transaction | 147 |
| 5.2 | SPI Interface (ROM Load Unit) | 148 |
| 5.2.1 | Accesses to a SPI EEPROM | 149 |
| 5.2.2 | SPI Read Sequence | 149 |
| 5.2.3 | SPI Write Sequence | 150 |
| 5.3 | Local Microprocessor Interface | 151 |
| 5.3.1 | Intel Mode | 152 |
| 5.3.1.1 | Slave Mode | 152 |
| 5.3.1.2 | Master Mode | 152 |
| 5.3.2 | Motorola Mode | 155 |
| 5.3.2.1 | Slave Mode | 155 |
| 5.3.2.2 | Master Mode | 155 |
| 5.4 | Serial Line Interface | 158 |
| 5.5 | JTAG Interface | 160 |
| 6 | Channel Programming / Reprogramming Concept | 162 |
| 6.1 | Channel Commands | 163 |
| 6.2 | Transmit Channel Commands | 163 |
| 6.3 | Receive Channel Commands | 165 |
| 7 | Reset and Initialization procedure | 168 |
| 7.1 | Chip Initialization | 168 |
| 7.2 | Mode Initialization | 169 |
| 8 | Register Description | 170 |
| 8.1 | Register Overview | 170 |
| 8.1.1 | PCI Configuration Register Set (Direct Access) | 170 |
| 8.1.2 | PCI Slave Register Set (Direct Access) | 172 |
| 8.1.3 | PCI and Local Bus Register Set (Direct Access) | 174 |
| 8.1.4 | Transmit T1/E1 Framer Registers (Indirect Access) | 179 |
| 8.1.5 | Receive T1/E1 Framer Registers (Indirect Access) | 180 |
| 8.1.6 | Facility Data Link Registers (Indirect Access) | 181 |
| 8.2 | Detailed Register Description | 182 |
| 8.2.1 | PCI Configuration Register | 182 |
| 8.2.2 | PCI Slave Register | 197 |
| 8.2.3 | PCI and Local Bus Slave Register Set | 246 |

| | | |
|---------|---|-----|
| 8.2.3.1 | M13 Transmit Registers | 262 |
| 8.2.3.2 | DS2 Control and Status Registers | 292 |
| 8.2.4 | Test Unit Registers | 307 |
| 8.2.5 | Transmit Framer Register | 325 |
| 8.2.6 | Receive Framer Registers | 335 |
| 8.2.7 | Facility Data Link Registers | 359 |
| 9 | Electrical Characteristics | 388 |
| 9.1 | Important Electrical Requirements | 388 |
| 9.2 | Absolute Maximum Ratings | 388 |
| 9.3 | DC Characteristics | 388 |
| 9.4 | AC Characteristics | 390 |
| 9.4.1 | PCI Bus Interface Timing | 391 |
| 9.4.2 | SPI Interface Timing | 393 |
| 9.4.3 | Local Microprocessor Interface Timing | 394 |
| 9.4.3.1 | Intel Bus Interface Timing (Slave Mode) | 394 |
| 9.4.3.2 | Intel Bus Interface Timing (Master Mode) | 396 |
| 9.4.3.3 | Motorola Bus Interface Timing (Slave Mode) | 399 |
| 9.4.3.4 | Motorola Bus Interface Timing (Master Mode) | 401 |
| 9.4.4 | Serial Interface Timing | 405 |
| 9.4.4.1 | DS3 Serial Interface Timing | 405 |
| 9.4.4.2 | Overhead Bit Timing | 409 |
| 9.4.4.3 | Stuff Bit Timing | 411 |
| 9.4.4.4 | T1/E1 Tributary Timing | 412 |
| 9.4.4.5 | Test Port Timing | 414 |
| 9.4.5 | JTAG Interface Timing | 417 |
| 9.4.6 | Reset Timing | 418 |
| 10 | Package Outline | 419 |

| | | |
|-------------|--|-----|
| Figure 1-1 | MUNICH256FM Logic Symbol | 19 |
| Figure 1-2 | System Integration of the MUNICH256FM | 20 |
| Figure 3-1 | MUNICH256FM Block Diagram | 42 |
| Figure 4-1 | Port configuration in M13 mode | 47 |
| Figure 4-2 | Local Port Loops in M13 mode | 48 |
| Figure 4-3 | Remote Line Loops | 49 |
| Figure 4-4 | Test Breakout. | 50 |
| Figure 4-5 | Time slot Assignment in Channelized Modes | 52 |
| Figure 4-6 | Descriptor Structure | 54 |
| Figure 4-7 | Receive Buffer Thresholds. | 68 |
| Figure 4-8 | Transmit Buffer Thresholds | 69 |
| Figure 4-9 | HDLC Frame Format | 71 |
| Figure 4-10 | Bit Synchronous PPP with HDLC Framing Structure. | 72 |
| Figure 4-11 | CRC-4 Multiframe Alignment Recovery Algorithms | 91 |
| Figure 4-12 | Interrupt Driven Reception Sequence Example. | 103 |
| Figure 4-13 | Interrupt Driven Transmit Sequence Example | 104 |
| Figure 4-14 | Test Unit Access Points | 118 |
| Figure 4-15 | Pattern Generator | 119 |
| Figure 4-16 | Mailbox Structure. | 120 |
| Figure 4-17 | Layer Two Interrupts (Channel, command, port and system interrupts | 123 |
| Figure 4-18 | Interrupt Queue Structure in System Memory | 124 |
| Figure 4-19 | Framer, M13 and Facility Data Link and Mailbox Interrupt Notification | 136 |
| Figure 5-1 | PCI Read Transaction | 147 |
| Figure 5-2 | PCI Write Transaction | 148 |
| Figure 5-3 | SPI Read Sequence | 150 |
| Figure 5-4 | SPI Write Sequence. | 150 |
| Figure 5-5 | Intel Bus Mode | 153 |
| Figure 5-6 | Intel Bus Arbitration | 153 |
| Figure 5-7 | Motorola Bus Mode | 156 |
| Figure 5-8 | Motorola Bus Arbitration | 156 |
| Figure 5-9 | Receive Overhead Access. | 158 |
| Figure 5-10 | Transmit Overhead Access | 159 |
| Figure 5-11 | Block Diagram of Test Access Port and Boundary Scan Unit | 160 |
| Figure 9-1 | Input/Output Waveform for AC Tests. | 390 |
| Figure 9-2 | PCI Clock Cycle Timing | 391 |
| Figure 9-3 | PCI Input Timing Measurement Conditions | 391 |
| Figure 9-4 | PCI Output Timing Measurement Conditions | 392 |
| Figure 9-5 | SPI Interface Timing | 393 |
| Figure 9-6 | Intel Read Cycle Timing (Slave Mode) | 394 |
| Figure 9-7 | Intel Write Cycle Timing (Slave Mode). | 394 |
| Figure 9-8 | Intel Read Cycle Timing (Master Mode, LRDY controlled) | 396 |
| Figure 9-9 | Intel Write Cycle Timing (Master Mode, LRDY controlled). | 396 |

| | | |
|-------------|---|-----|
| Figure 9-10 | Intel Read Cycle Timing (Master Mode, Wait state controlled) | 397 |
| Figure 9-11 | Intel Write Cycle Timing (Master Mode, Wait state controlled) | 397 |
| Figure 9-12 | Intel Bus Arbitration Timing | 398 |
| Figure 9-13 | Motorola Read Cycle Timing (Slave Mode) | 399 |
| Figure 9-14 | Motorola Write Cycle Timing (Slave Mode) | 399 |
| Figure 9-15 | Motorola Read Cycle Timing (Master Mode, <u>LDTACK</u> controlled) . . . | 401 |
| Figure 9-16 | Motorola Write Cycle Timing (Master Mode, <u>LDTACK</u> controlled) . . . | 401 |
| Figure 9-17 | Motorola Read Cycle Timing (Master Mode, Wait state controlled) . . . | 402 |
| Figure 9-18 | Motorola Write Cycle Timing (Master Mode, Wait state controlled) . . . | 402 |
| Figure 9-19 | Motorola Bus Arbitration Timing | 403 |
| Figure 9-20 | Clock Input Timing | 405 |
| Figure 9-21 | DS3 Transmit Cycle Timing | 406 |
| Figure 9-22 | DS3 Transmit Data Timing | 406 |
| Figure 9-23 | DS3 Receive Cycle Timing | 407 |
| Figure 9-24 | DS3 Status Signal Timing | 408 |
| Figure 9-25 | DS3 Transmit Overhead Timing | 409 |
| Figure 9-26 | DS3 Transmit Overhead Synchronization Timing | 409 |
| Figure 9-27 | DS3 Receive Overhead Timing | 410 |
| Figure 9-28 | DS3 Transmit Stuff Bit Timing | 411 |
| Figure 9-29 | DS3 Receive Stuff Bit Timing | 411 |
| Figure 9-30 | T1/E1 Tributary Clock Input Timing | 412 |
| Figure 9-31 | T1/E1 Tributary Synchronization Timing | 413 |
| Figure 9-32 | T1/E1 Test Transmit Clock Timing | 414 |
| Figure 9-33 | T1/E1 Test Transmit Data Timing | 415 |
| Figure 9-34 | T1/E1 Test Receive Clock Timing | 415 |
| Figure 9-35 | T1/E1 Test Receive Data Timing | 416 |
| Figure 9-36 | JTAG Interface Timing | 417 |
| Figure 9-37 | Reset Timing | 418 |

| | | |
|------------|---|-----|
| Table 4-1 | Receive Descriptor Structure | 55 |
| Table 4-2 | Transmit Descriptor Structure | 61 |
| Table 4-3 | Example for little/big Endian with BNO = 3 | 66 |
| Table 4-4 | Example for little big Endian with BNO = 7 | 66 |
| Table 4-5 | 4-Frame Multiframe Structure. | 74 |
| Table 4-6 | ESF Multiframe Structure. | 75 |
| Table 4-7 | SF Multiframe Structure | 80 |
| Table 4-8 | Allocation of Bits 1 to 8 of Time slot 0 | 86 |
| Table 4-9 | CRC-4 Multiframe Structure. | 88 |
| Table 4-10 | Summary of Alarm Detection and Alarm Release | 94 |
| Table 4-11 | M12 multiplex format | 105 |
| Table 4-12 | ITU-T G.747 format | 108 |
| Table 4-13 | M23 multiplex format | 110 |
| Table 4-14 | C-bit parity format | 113 |
| Table 4-15 | Interrupt Vector Structure. | 124 |
| Table 5-1 | Correspondence between PCI memory space and chip select | 151 |
| Table 5-2 | C/BE to LA/LBHE mapping in Intel bus mode (8 bit port mode) | 154 |
| Table 5-3 | C/BE to LA/LBHE mapping in Intel bus mode (16 bit port mode) | 154 |
| Table 5-4 | C/BE to LA/LSIZE0 mapping in Motorola bus mode (8 bit port mode) | 157 |
| Table 5-5 | C/BE to LA/LSIZE0 mapping in Motorola bus mode (16 bit port mode) | 157 |
| Table 6-1 | Channel Specification Registers and Channel Commands | 162 |
| Table 8-1 | PCI Configuration Register Set | 170 |
| Table 8-2 | PCI Slave Register Set | 172 |
| Table 8-3 | PCI and Local Bus Slave Register Set | 174 |
| Table 8-4 | Transmit T1/E1 Framers Registers | 179 |
| Table 8-5 | Receive T1/E1 Framers Registers. | 180 |
| Table 8-6 | Facility Data Link Registers | 181 |
| Table 8-7 | Threshold Codings. | 208 |
| Table 8-8 | Signalling Controller Transmit Commands | 373 |
| Table 9-1 | Absolute Maximum Ratings | 388 |
| Table 9-2 | DC Characteristics. | 388 |
| Table 9-3 | DC Characteristics (Non-PCI Interface Pins). | 389 |
| Table 9-4 | DC Characteristics (PCI Interface Pins). | 389 |
| Table 9-5 | PCI Clock Characteristics | 391 |
| Table 9-6 | PCI Interface Signal Characteristics | 392 |
| Table 9-7 | SPI Interface Timing | 393 |
| Table 9-8 | Intel Bus Interface Timing | 395 |
| Table 9-9 | Intel Bus Interface Timing (Master Mode) | 398 |
| Table 9-10 | Motorola Bus Interface Timing. | 400 |
| Table 9-11 | Motorola Bus Interface Timing (Master Mode). | 403 |
| Table 9-12 | Clock Input Timing | 405 |
| Table 9-13 | DS3 Transmit Cycle Timing | 406 |

| | | |
|------------|--|-----|
| Table 9-14 | DS3 Receive Cycle Timing | 407 |
| Table 9-15 | DS3 Status Signal Timing | 408 |
| Table 9-16 | DS3 Transmit Overhead Timing | 409 |
| Table 9-17 | DS3 Receive Overhead Timing | 410 |
| Table 9-18 | DS3 Transmit Stuff Timing | 411 |
| Table 9-19 | DS3 Receive Stuff Bit Timing | 411 |
| Table 9-20 | T1/E1 Tributary Clock Input Timing | 412 |
| Table 9-21 | T1/E1 Tributary Synchronization Timing | 413 |
| Table 9-22 | T1/E1 Test Transmit Clock Timing | 414 |
| Table 9-23 | T1/E1 Test Transmit Data Timing | 415 |
| Table 9-24 | T1/E1 Test Receive Clock Timing | 415 |
| Table 9-25 | Test T1/E1 Receive Data Timing | 416 |
| Table 9-26 | JTAG Interface Timing | 417 |
| Table 9-27 | Reset Timing | 418 |

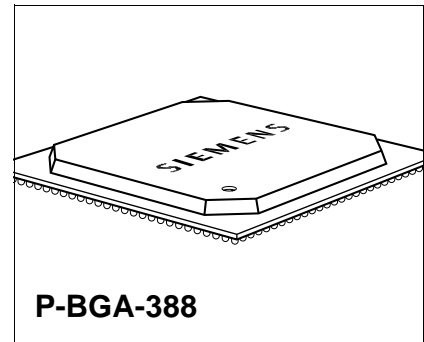
Multichannel Network Interface Controller for HDLC/PPP PEB 20256M E MUNICH256FM

Version 1.1

CMOS

1 MUNICH256FM Overview

The MUNICH256FM is a highly integrated protocol controller that implements HDLC, PPP and transparent (TMA) protocol processing for 256 channels as well as frame alignment for up to 28 T1 signals or 21 E1 signals. An integrated M13 multiplexer together with a DS3 framer concentrates the data links for direct connection to a DS3 line interface unit. Optionally the device supports unchannelized DS3 applications. An



internal bit error rate tester can be attached to different test points and provides flexible PRBS and fixed pattern tests. An on-chip data management unit is optimized to transfer data packets via a PCI interface by minimizing the bus load.

Note: The MUNICH256FM does not contain DS3 Line Interface Units.

1.1 General Features

- Protocol processing on a channelized or unchannelized DS3 link for frame relay or router applications
- Direct connection to DS3 line interface unit or DS3 to STS-1 mapper
- Support of 256 bidirectional channels, which can be assigned arbitrarily to a maximum of 28 links, for HDLC, PPP or transparent mode (TMA) processing
- Concatenation of any, not necessarily consecutive, time slots to logical channels on each physical link. Supports DS0, fractional T1/E1 or T1/E1 channels
- Provides 32kB data buffer in transmit direction and 12kB data buffer in receive direction
- Integrates 28T1/21E1 framers (frame alignment function) and 28T1/21E1 signalling controllers
- Integrates a DS2/DS3 multiplexer and framer
- Remote loopbacks selectable for either DS3 signal, DS2 signal or T1/E1 signal/payload

| Type | Package |
|--------------|-----------|
| PEB 20256M E | P-BGA-388 |

- System interface is a PCI 32 bit, 66 MHz Rev. 2.1 compliant bus interface, which supports configuration of subsystem ID / subsystem vendor ID via a serial EEPROM interface. PCI bus interface can be operated in the range of 33 MHz to 66 MHz
- Integrates a local microprocessor master and slave interface (demultiplexed 16 bit address and data bus in Intel mode or Motorola mode) which allows access to the local bus via the PCI bus or which can communicate with a PCI host processor through an on-chip mailbox
- For debugging purposes optional access to the framer and signalling controller functions via the PCI interface
- JTAG boundary scan according to IEEE1149.1 (5 pins).
- 0.25 μm , 2.5V core technology
- I/Os are 3.3V tolerant and have 3.3V driving capability
- Package P-BGA 388 (35mm x 35mm; pitch 1.27mm)
- Full scan path and BIST of on-chip RAMs for production test
- Performance: 45Mbit/s (DS3) throughput per direction
- Estimated power consumption: 2W
- Also available as device with extended temperature range -40..+85°C

1.1.1 M12 Multiplexer and DS2 Framer

- Multiplexing/Demultiplexing of four asynchronous DS1 bit streams into/from M13 asynchronous format
- Multiplexing/Demultiplexing of 3 E1 signals into/from ITU G.747 compliant DS2 signal.
- DS2 line loopback detection/generation
- Framing according to ANSI T1.107, T1.107a or ITU-T G.747
- Insertion and extraction of X-bit
- Insertion and Extraction of alarms (remote alarm, AIS)
- Detection of AIS in presence of BER 10^{-3}
- Alarm and performance monitoring (framing bit errors, parity errors)
- Reframe time below 7ms (TR-TSY-000009) for DS2 format and below 1 ms for ITU G.747 format
- Bit Stuffing/Destuffing in M12 multiplex format or C-bit parity format

1.1.2 M23 Multiplexer and DS3 Framer

- Multiplexing/demultiplexing of seven DS2 into/from M13 asynchronous format according to ANSI T1.107, ANSI T1.107a
- Multiplexing/demultiplexing of seven DS2 into/from C-bit parity format according to ANSI T1.107, ITU-T G.704
- DS3 framing according to ANSI T1.107, T1.107a, ITU-T G.704
- Support of unipolar and B3ZS encoded signals
- Provides access to the DS3 overhead bits and the DS3 stuffing bits via a serial clock and data interface (overhead interface)

- Insertion and Extraction of alarms according to ANSI T1.404 (remote alarm, AIS, far end receive failure)
- Supports HDLC (Path Maintenance Data Link) and bit oriented message mode (Far End Alarm and Control Channel) in C-bit parity mode. An integrated signalling controller provides 2x32 byte deep FIFO's for each direction of both channels
- Detection of AIS and idle signal in presence of BER 10^{-3}
- Detection of excessive zeroes and LOS
- Alarm and performance monitoring with 16-bit counters for line code violations, excessive zeroes, parity error (P-bit), framing errors (F-bit errors with or without M-bit errors, far end block error (FEBE-bit) and CP-bit errors.
- Automatic insertion of severely errored frame and AIS defect indication

1.1.3 Frame Alignment T1 Features

- Frame alignment/synthesis for 1544 kbit/s according to ITU-T G.704
- Supports T1 frame alignment for F4, SF (F12) and ESF (F24) mode
- Error checking via CRC-6 procedures according to ITU-T G.706
- Performance monitor: 16 bit counter for CRC, framing errors, loss of frame alignment, loss of signal AIS
- Insertion and extraction of alarms (AIS, Remote (Yellow) Alarm)
- Detection of LOS (Red Alarm)
- Pseudo-random bit sequence generator and monitor for one logical channel according to ITU-T O.151
- Programmable in-band loop code detection/generation according to TR 62411

1.1.4 Signaling Controller T1 Features

- FDL-channel protocol for ESF format according to ANSI T1.403 specification or according to AT&T TR54016
- Supports HDLC mode with address recognition
- Supports BOM mode
- FIFO Buffers (64 bytes deep) for efficient transfer of data packets

1.1.5 Frame Alignment E1 Features

- Frame alignment/synthesis for 2048 kbit/s according to ITU-T G.704
- Programmable formats: Doubleframe, CRC-4 Multiframe
Selectable conditions for recover / loss of frame alignment
- CRC-4 to Non-CRC-4 Interworking of ITU-T G.706 Annex B
- Error checking via CRC-4 procedures according to ITU-T G.706
- Performance monitor: 16 bit counter for CRC-, framing errors, error monitoring via E-bit and S_{a6} bit
- Insertion and extraction of alarms (AIS, Remote (Yellow) Alarm, ...)
- Pseudo-random bit sequence (PRBS) generator and monitor for one logical channel

- Programmable in-band loop code detection / generation according to TR 62411

1.1.6 Signaling Controller E1 Features

- HDLC controller with address recognition and programmable preamble
- Time slot 0 S_{a8-4} HDLC handling via FIFOs
- HDLC access to any S_a -bit combination
- FIFO Buffers (64 byte deep) for efficient transfer of data packets

1.1.7 Bit Error Rate Tester

- User specified PRBS/Fixed Pattern with programmable length of 1 to 32 bits
- Optional Bit Inversion
- Two error insertion modes: Single or programmable bit rates
- Optional zero suppression
- 32-bit counters for errors and received bits
- Programmable bit intervals for receive measurements

1.2 Logic Symbol

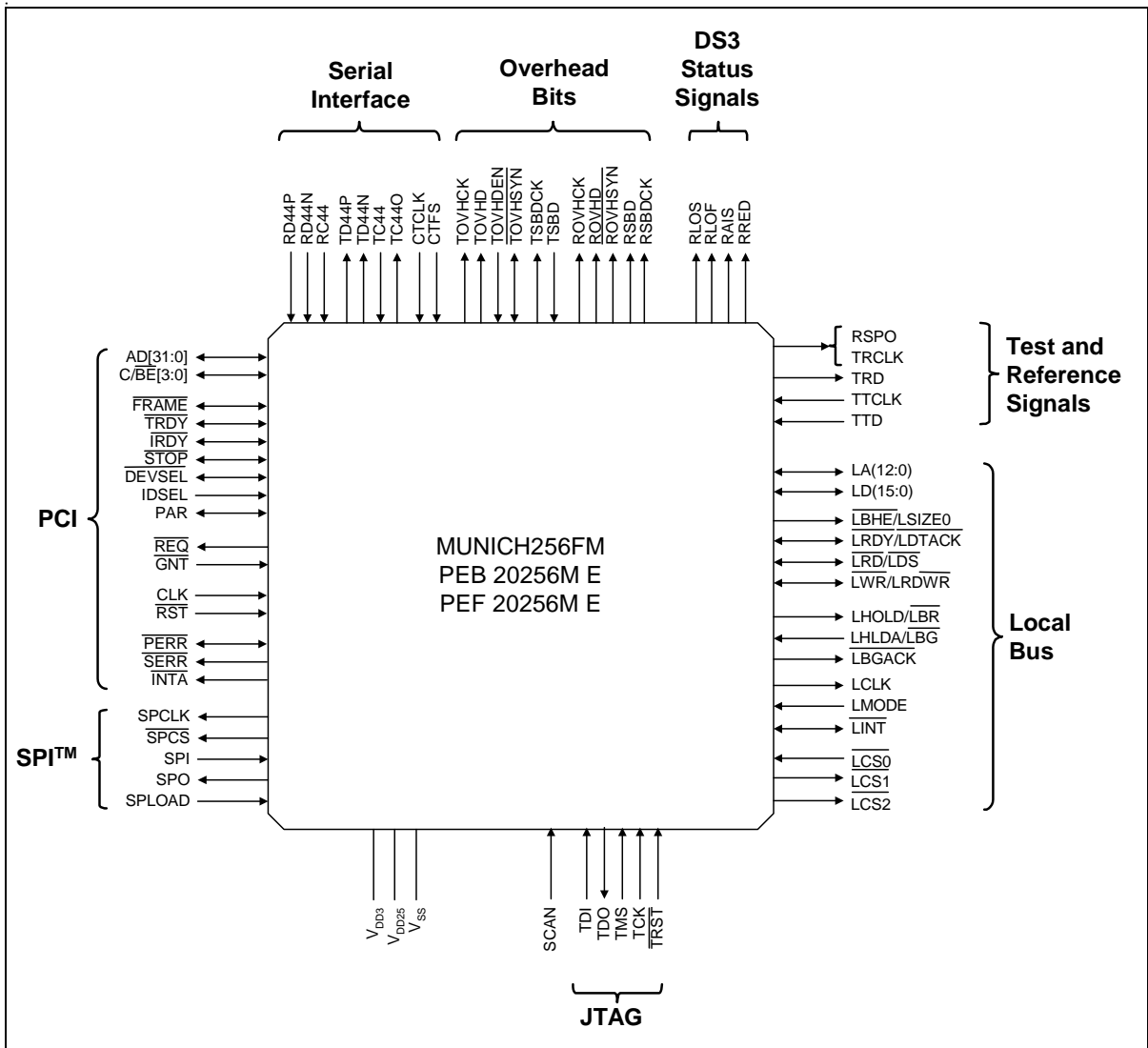


Figure 1-1 MUNICH256FM Logic Symbol

1.3 General System Integration

The MUNICH256FM provides the HDLC/PPP protocol handling, T1/E1 framing and signalling functions, an integrated M13 multiplexer and a DS3 framer. The line interface of the MUNICH256FM directly connects to a DS3 line interface unit. Protocol data is transferred to the packet RAM via the PCI bus and handled (e.g. for layer3 protocol handling) by the line card processor. An external processor provides control of the integrated T1/E1 framer, M13 multiplexer, DS3 framer and the signalling channels. A mailbox allows the transfer of information between both CPUs.

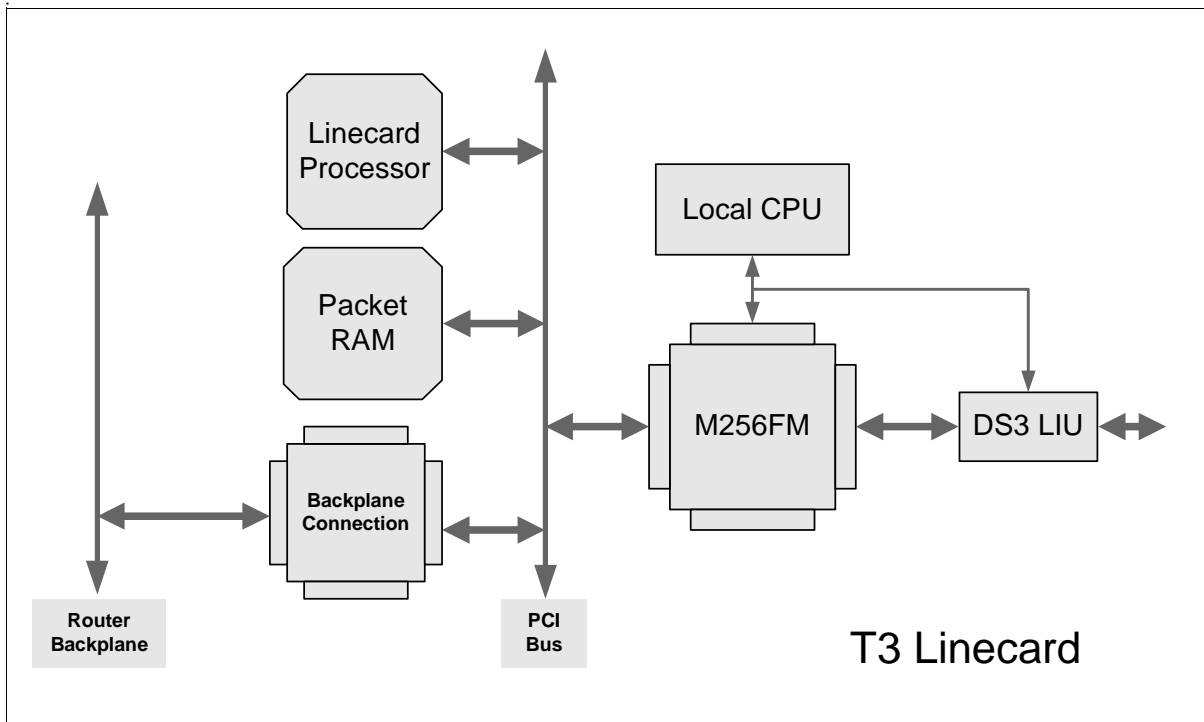


Figure 1-2 System Integration of the MUNICH256FM

2 Pin Description

Signal Type Definitions:

The following signal type definitions are partly taken from the PCI Specification Rev. 2. 1:

- I** *Input* is a standard input- only signal.
- O** *Totem Pole Output* is a standard active driver.
- t/s, I/O** *Tri-State* or *I/O* is a bidirectional, tri-state input/output pin.
- s/t/s** *Sustained Tri-State* is an active low tri-state signal owned and driven by one and only agent at a time. The agent that drives an s/t/s pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. A pullup is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.
- o/d** *Open Drain* allows multiple devices to share a line as a wire-OR. A pull-up is required to sustain the inactive state until another agent drives it, and must be provided by the central resource.

Signal Name Conventions:

- NCn** *No-connect Pin n*
- Such pins are not bonded with the silicon. Although any potential at these pins will not impact the device it is recommended to leave them unconnected. No-connect pins might be used for additional functionality in later versions of the device. Leaving them unconnected will guarantee hardware compatibility to later device versions.
- Reserved** Reserved pins are for vendor specific use only and should be connected as recommended to guarantee normal operation.

Note: The signal type definition specifies the functional usage of a pin. This does not reflect necessarily the implementation of a pin, e.g. a pin defined of signal type 'Input' may be implemented with a bidirectional pad.

2.1 PCI Bus Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|----------|-------------------------|---|
| T3, T4, U1, U3, V2, W1, W2, V4, AA2, W4, AC1, AB2, Y3, Y4, AD1, AC2, AC8, AE6, AD8, AF6, AC9, AE8, AF7, AD10, AC11, AF8, AF10, AD11, AC12, AE11, AD12, AF11 | AD(31:0) | t/s | <p>Address/Data Bus</p> <p>A bus transaction consists of an address phase followed by one or more data phases.</p> <p>When the MUNICH256FM is the bus master, AD(31:0) are outputs in the address phase of a transaction. During the data phases, AD(31:0) remain outputs for write transactions, and become inputs for read transactions.</p> <p>When the MUNICH256FM is bus slave, AD(31:0) are inputs in the address phase of a transaction. During the data phases, AD(31:0) remain inputs for write transactions, and become outputs for read transactions.</p> <p>AD(31:0) are tri-state when the MUNICH256FM is not involved in the current transaction.</p> <p>AD(31:0) are updated and sampled on the rising edge of CLK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|----------------------|------------------------|-------------------------|---|
| V3, AA4, AD7, AE9 | $\overline{C/BE}(3:0)$ | t/s | <p>Command/Byte Enable</p> <p>During the address phase of a transaction, $\overline{C/BE}(3:0)$ define the bus command. During the data phase, $\overline{C/BE}(3:0)$ are used as byte enable lines. The byte enable lines are valid for the entire data phase and determine which byte lanes carry meaningful data. $\overline{C/BE}(0)$ applies to byte 0 (LSB) and $\overline{C/BE}(3)$ applies to byte 3 (MSB).</p> <p>When the MUNICH256FM is bus master, $\overline{C/BE}(3:0)$ are outputs.</p> <p>When the MUNICH256FM is bus slave, $\overline{C/BE}(3:0)$ are inputs.</p> <p>$\overline{C/BE}(3:0)$ are tri-stated when the MUNICH256FM is not involved in the current transaction.</p> <p>$\overline{C/BE}(3:0)$ are updated and sampled on the rising edge of CLK.</p> |
| AF4 | PAR | t/s | <p>Parity</p> <p>PAR is even parity across AD(31:0) and $\overline{C/BE}(3:0)$. PAR is stable and valid one clock after the address phase. PAR has the same timing as AD(31:0) but delayed by one clock.</p> <p>When the MUNICH256FM is Master, PAR is output during address phase and write data phases and input during read data phase. When the MUNICH256FM is Slave, PAR is output during read data phase and input during write data phase. PAR is tri-stated when the MUNICH256FM is not involved in the current transaction.</p> <p>Parity errors detected by the MUNICH256FM are indicated on PERR output.</p> <p>PAR is updated and sampled on the rising edge of CLK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|---------------------------|-------------------------|--|
| AB3 | $\overline{\text{FRAME}}$ | s/t/s | <p>Frame $\overline{\text{FRAME}}$ indicates the beginning and end of an access. $\overline{\text{FRAME}}$ is asserted to indicate a bus transaction is beginning. While $\overline{\text{FRAME}}$ is asserted, data transfers continue. When $\overline{\text{FRAME}}$ is deasserted, the transaction is in the final phase. When the MUNICH256FM is bus master, $\overline{\text{FRAME}}$ is an output. When the MUNICH256FM is bus slave, $\overline{\text{FRAME}}$ is an input. $\overline{\text{FRAME}}$ is tri-stated when the MUNICH256FM is not involved in the current transaction. $\overline{\text{FRAME}}$ is updated and sampled on the rising edge of CLK.</p> |
| AC6 | $\overline{\text{IRDY}}$ | s/t/s | <p>Initiator Ready $\overline{\text{IRDY}}$ indicates the bus master's ability to complete the current data phase of the transaction. It is used in conjunction with $\overline{\text{TRDY}}$. A data phase is completed on any clock where both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are sampled asserted. During a write, $\overline{\text{IRDY}}$ indicates that valid data is present on AD(31:0). During a read, it indicates the master is prepared to accept data. Wait cycles are inserted until both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted together. When the MUNICH256FM is bus master, $\overline{\text{IRDY}}$ is an output. When the MUNICH256FM is bus slave, $\overline{\text{IRDY}}$ is an input. $\overline{\text{IRDY}}$ is tri-stated, when the MUNICH256FM is not involved in the current transaction. $\overline{\text{IRDY}}$ is updated and sampled on the rising edge of CLK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|--------------------------|-------------------------|---|
| AD5 | $\overline{\text{TRDY}}$ | s/t/s | <p>Target Ready</p> <p>$\overline{\text{TRDY}}$ indicates a slave's ability to complete the current data phase of the transaction. During a read, $\overline{\text{TRDY}}$ indicates that valid data is present on AD(31:0). During a write, it indicates the target is prepared to accept data.</p> <p>When the MUNICH256FM is Master, $\overline{\text{TRDY}}$ is an input. When the MUNICH256FM is Slave, $\overline{\text{TRDY}}$ is an output. $\overline{\text{TRDY}}$ is tri-stated, when the MUNICH256FM is not involved in the current transaction.</p> <p>$\overline{\text{TRDY}}$ is updated and sampled on the rising edge of CLK.</p> |
| AF3 | $\overline{\text{STOP}}$ | s/t/s | <p>Stop</p> <p>$\overline{\text{STOP}}$ is used by a slave to request the current master to stop the current bus transaction.</p> <p>When the MUNICH256FM is bus master, $\overline{\text{STOP}}$ is an input. When the MUNICH256FM is bus slave, $\overline{\text{STOP}}$ is an output. $\overline{\text{STOP}}$ is tri-stated, when the MUNICH256FM is not involved in the current transaction.</p> <p>$\overline{\text{STOP}}$ is updated and sampled on the rising edge of CLK.</p> |
| AA1 | IDSEL | I | <p>Initialization Device Select</p> <p>When the MUNICH256FM is slave in a transaction, where $\overline{\text{IDSEL}}$ is active in the address phase and C/BE(3:0) indicates an configuration read or write, the MUNICH256FM assumes a read or write to a configuration register. In response, the MUNICH256FM asserts $\overline{\text{DEVSEL}}$ during the subsequent CLK cycle.</p> <p>IDSEL is sampled on the rising edge of CLK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|----------------------------|-------------------------|---|
| AE4 | $\overline{\text{DEVSEL}}$ | s/t/s | <p>Device Select</p> <p>When activated by a slave, it indicates to the current bus master that the slave has decoded its address as the target of the current transaction. If no bus slave activates $\overline{\text{DEVSEL}}$ within six bus CLK cycles, the master should abort the transaction.</p> <p>When the MUNICH256FM is bus master, $\overline{\text{DEVSEL}}$ is input. If $\overline{\text{DEVSEL}}$ is not activated within six clock cycles after an address is output on AD(31:0), the MUNICH256FM aborts the transaction.</p> <p>When the MUNICH256FM is bus slave, $\overline{\text{DEVSEL}}$ is output. $\overline{\text{DEVSEL}}$ is tri-stated, when the MUNICH256FM is not involved in the current transaction.</p> |
| AC7 | $\overline{\text{PERR}}$ | s/t/s | <p>Parity Error</p> <p>When activated, indicates a parity error over the AD(31:0) and C/BE(3:0) signals (compared to the PAR input). It has a delay of two CLK cycles with respect to AD and C/BE(3:0) (i.e., it is valid for the cycle immediately following the corresponding PAR cycle).</p> <p>$\overline{\text{PERR}}$ is asserted relative to the rising edge of CLK.</p> |
| AE5 | $\overline{\text{SERR}}$ | o/d | <p>System Error</p> <p>The MUNICH256FM asserts this signal to indicate an address parity error and report a fatal system error.</p> <p>$\overline{\text{SERR}}$ is an open drain output activated on the rising edge of CLK.</p> |
| T2 | $\overline{\text{REQ}}$ | t/s | <p>Request</p> <p>Used by the MUNICH256FM to request control of the PCI bus. It is tri-state during reset.</p> <p>$\overline{\text{REQ}}$ is activated on the rising edge of CLK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|--------------------------|-------------------------|--|
| T1 | $\overline{\text{GNT}}$ | I | <p>Grant</p> <p>This signal is asserted by the arbiter to grant control of the PCI to the MUNICH256FM in response to a bus request via REQ. After GNT is asserted, the MUNICH256FM will begin a bus transaction only after the current bus Master has deasserted the $\overline{\text{FRAME}}$ signal.</p> <p>$\overline{\text{GNT}}$ is sampled on the rising edge of CLK.</p> |
| R4 | CLK | I | <p>Clock</p> <p>Provides timing for all PCI transactions. Most PCI signals are sampled or output relative to the rising edge of CLK. The PCI clock is used as internal system clock. The maximum CLK frequency is 66 MHz.</p> |
| R3 | $\overline{\text{RST}}$ | I | <p>Reset</p> <p>An active $\overline{\text{RST}}$ signal brings all PCI registers, sequencers and signals into a consistent state. All PCI output signals are driven to high impedance.</p> |
| AC13 | $\overline{\text{INTA}}$ | o/d | <p>Interrupt Request</p> <p>When an interrupt status is active and unmasked, the MUNICH256FM activates this open-drain output.</p> |

2.2 SPI Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|--------------------------|-------------------------|--|
| P2 | SPI | I | SPI Serial Input SPI is a data input pin, where data coming from an external EEPROM is shifted in. SPI is sampled on the rising edge of SPCLK. A pull-up resistor is recommended if the SPI interface is not used. |
| P1 | SPO | O | SPI Serial Output SPO is a push/pull serial data output pin. Opcodes, byte addresses and data is updated on the falling edge of SPCLK. It is tri-state during reset. |
| N4 | SPCLK | O | SPI Clock Signal SPCLK controls the serial bus timing of the SPI bus. SPCLK is derived from the PCI bus clock with a frequency of 1/78 of the PCI bus clock. It is tri-state during reset. |
| N3 | $\overline{\text{SPCS}}$ | O | SPI Chip Select $\overline{\text{SPCS}}$ is used to select an external EEPROM. It is tri-state during reset. |
| P4 | SPLOAD | I | Enable SPI Load Functionality Connecting SPLOAD to V_{DD3} enables the SPI bus after reset. In this case parts of the PCI configuration space can be configured via an external EEPROM. |

2.3 Local Microprocessor Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---|--------------------------|-------------------------|--|
| W24 | LMODE | I | <p>Local Bus Mode</p> <p>By connecting this pin to either V_{SS} or V_{DD3} the bus interface can be adapted to either Intel or Motorola environment.</p> <p>LMODE = V_{SS} selects Intel bus mode.</p> <p>LMODE = V_{DD3} selects Motorola bus mode.</p> |
| Y24 | LCLK | O | <p>Local Clock</p> <p>Reference output clock derived from the PCI clock.</p> |
| AE13, AF13, AF14, AE14, AF16, AC14, AD15, AE16, AF17, AC15, AD16, AF19, AE18 | LA(12:0) | I/O | <p>Address bus</p> <p>These input address lines select one of the internal registers for read or write access.</p> <p><i>Note: Only LA(7:0) are evaluated during read/write accesses to the MUNICH256FM.</i></p> <p>In local bus master mode the address lines are output. If local bus master functionality is disabled these pins are input only.</p> |
| AC16, AD17, AF20, AE19, AF21, AC18, AD19, AE21, AD20, AC19, AF23, AE24, AF25, AE26, AD25, AB23 | LD(15:0) | I/O | <p>Data Bus</p> <p>Bidirectional tri-state data lines.</p> |
| Y23 | $\overline{\text{LCS0}}$ | I | <p>Chip Select</p> <p>This active low signal selects the MUNICH256FM as bus slave for read/write operations.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|---------------------------------|-------------------------|--|
| AC24 | $\overline{\text{LRD}}$ | I/O | Read (Intel Bus Mode) This active low signal selects a read transaction. |
| | or $\overline{\text{LDS}}$ | I/O | Data strobe (Motorola Bus Mode) This active low signal indicates that valid data has to be placed on the data bus (read cycle) or that valid data has been placed on the data bus (write cycle). |
| AB24 | $\overline{\text{LWR}}$ | I/O | Write Enable (Intel Bus Mode) This active low signal selects a write cycle. |
| | or $\overline{\text{LRDWR}}$ | I/O | Read Write Signal (Motorola Bus Mode) This input signal distinguishes write from read operations. |
| AA23 | $\overline{\text{LRDY}}$ | I/O | Ready (Intel bus mode) This signal indicates that the current bus cycle is <u>complete</u> . The MUNICH256FM asserts $\overline{\text{LRDY}}$ during a read cycle if valid output data has been placed on the data bus. In write direction $\overline{\text{LRDY}}$ will be asserted when input data has been latched. |
| | or $\overline{\text{DTACK}}$ | I/O | In local bus <u>master</u> mode MUNICH256FM evaluates $\overline{\text{LRDY}}$ to finish a transaction. Data Transfer Acknowledge (Motorola bus mode) This active low input indicates that a data transfer may be performed. During a read cycle data <u>becomes</u> valid at the falling edge of $\overline{\text{DTACK}}$. The data is latched internally and the bus cycle is terminated. <u>During</u> a write cycle the falling edge of $\overline{\text{DTACK}}$ marks the latching of data and the bus cycle is terminated. |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|-----------|--|-------------------------|---|
| AC26 | $\overline{\text{LINT}}$ | I/od | <p>Interrupt Request This line indicates general interrupt requests of the layer one functions or the mailbox. The interrupt sources can be masked via registers. In local bus master mode the MUNICH256FM can monitor external interrupts indicated via $\overline{\text{LINT}}$.</p> |
| AC25, W23 | $\overline{\text{LCS2}}$, $\overline{\text{LCS1}}$ | O | <p>Chip Select 2, 1 These signals select external peripherals when MUNICH256FM is the local bus master. As long as the local bus master functionality is disabled these outputs are set to tri-state.</p> |
| AD13 | $\overline{\text{LBHE}}$ or LSIZE0 | O O | <p>Byte High Enable (Intel Bus Mode) In local bus master mode this signal indicates a data transfer on the upper byte of the data bus LD(15:8). This signal has no function in slave mode. When local bus master functionality is disabled this output is tri-state.</p> <p>Byte Access (Motorola Bus Mode) In local bus master mode this signal indicates byte transfers. This signal has no function when the MUNICH256FM is local bus slave. When local bus master functionality is disabled this output is tri-state.</p> |
| AA25 | LHOLD or $\overline{\text{LBR}}$ | O O | <p>Bus Request (Intel Bus Mode) This pin indicates a requests to become local bus master. When local bus master functionality is disabled this output is tri-state.</p> <p>Bus Request (Motorola Bus Mode) $\overline{\text{LBR}}$ indicates a request to become local bus master. When local bus master functionality is disabled this output is set to tri-state.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|-------------------------------|-------------------------|--|
| AB25 | LHLDA | I | Hold (Intel Bus Mode) LHLDA indicates that the external processor has released control of the local bus. |
| | or $\overline{\text{LBG}}$ | I | Bus Grant (Motorola Bus Mode) LBG indicates that the MUNICH256FM may access the local bus. |
| V23 | $\overline{\text{LBGACK}}$ | O | Bus Grant Acknowledge (Motorola Bus Mode) LBGACK is driven low when the MUNICH256FM has become bus master. When local bus master functionality is disabled this output is tri-state. |

2.4 Serial Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|-----------------------------|-------------------------|---|
| D12 | CTCLK | I | Common Transmit Clock CTCLK is the external transmit clock for the T1 or E1 tributaries configured in external timing mode. |
| A11 | CTFS | I | Common Transmit Frame Synchronization CTFS is used to synchronize the T1/E1 transmit lines, which are clocked with CTCLK in external timing mode. If not used CTFS should be connected to V_{SS} . |
| C15 | RSPO or TRCLK | O O | Regenerated Sync Pulse RSPO supports debugging of the on-chip T1/E1 framing function. If the T1/E1 framer achieved synchronization, the internal synchronization pulse of one selected T1/E1 framer can be monitored on RSPO. Test Receive Clock In serial test mode the receive clock of one selected T1/E1 interface is directly fed to this output. |
| M24 | TRD | O | Test Receive Data In serial test mode the incoming data stream of one T1/E1 tributary is directly fed to this output. Test receive data is updated on the falling edge of the TRCLK. |
| N26 | TTCLK | I | Test Transmit Clock In serial test mode this clock provides the clock reference for the tributary provided via TTD. |
| C12 | TTD | I | Test Transmit Data In serial test mode the data stream provided via TTD replaces the E1/T1 data stream of the selected tributary. TTD is sampled on the rising edge of the TTCLK. |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|-------------|-------------------------|--|
| C14 | TC44 | I | DS3 Transmit Clock Input This clock provides a reference clock for the DS3 interface. The frequency of this clock is nominally 44.736 MHz. |
| D14 | TC44O | O | DS3 Transmit Clock Output This output is a buffered version of the selected transmit clock which can be set to RC44 or TC44. |
| B16 | TD44 | O | DS3 Transmit Data This unipolar serial data output represents the DS3 signal. TD44 is updated on the falling or rising edge of TC44. |
| | or TD44P | O | DS3 Transmit Positive Pulse In dual-rail mode this pin represents the positive pulse of the B3ZS encoded DS3 signal. TD44P is updated on the falling edge or rising edge of TC44O. |
| C16 | TD44N | O | DS3 Transmit Negative Pulse In dual-rail mode this pin represents the negative pulse of the B3ZS encoded DS3 signal. TD44N is updated on the falling or rising edge of TC44O. |
| B14 | RC44 | I | DS3 Receive Clock Input The frequency of this clock is nominally 44.736 MHz. |
| D13 | RD44 | I | DS3 Receive Data This unipolar serial data input represents the DS3 signal. RD44 is sampled on the falling or rising edge of RC44. |
| | or RD44P | I | DS3 Receive Positive Pulse In dual-rail mode this pin represents the positive pulse of the B3ZS encoded DS3 signal. RD44P is sampled on the falling or rising edge of RC44. |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|----------------|---------------|---------------------------------|--|
| A14 | RD44N | I | DS3 Receive Negative Pulse In dual-rail mode this pin represents the negative pulse of the B3ZS encoded DS3 signal. RD44 is sampled on the falling or rising edge of RC44. |
| A21 | RRED | O | Received RED This signal is asserted whenever the DS3 receive framer is in RED alarm state. |
| B21 | RLOS | O | Received LOS This signal is asserted whenever the received DS3 bit stream contained at least 175 consecutive '0's. |
| D19 | RLOF | O | Receive LOF This signal is asserted whenever the DS3 receive framer is in 'Loss of frame' state. |
| C19 | RAIS | O | Received AIS This signal is asserted whenever the DS3 receive framer is in AIS state. |
| B8 | TOVHCK | O | Transmit Overhead Bit Clock This signal provides the bit clock for the DS3 overhead bits of the outgoing DS3 frame. TOVHCK is nominally a 526 kHz clock. |
| C8 | TOVHD | I | Transmit Overhead Data The overhead bits of the outgoing DS3 frame can be provided via TOVHD. Transmit overhead data is sampled on the rising edge of TOVHCK and those bits which are enabled by TOVHEN are inserted in the overhead bit positions of the DS3 frame. |
| D8 | TOVHEN | I | Enable Transmit Overhead Data The asserted TOVHEN signal marks the bits to be inserted in the DS3 frame. TOVHEN is sampled together with TOVHD on the rising edge of TOVHD. |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|-----------------------------|-------------------------|---|
| A8 | $\overline{\text{TOVHSYN}}$ | I/O | <p>Transmit Overhead Synchronization $\overline{\text{TOVHSYN}}$ provides the means to align TOVHD to the first M-frame of the DS3 signal. If operated in output mode $\overline{\text{TOVHSYN}}$ it is asserted when the X-bit of the 1st subframe of the DS3 overhead bits has to be inserted via TOVHD. $\overline{\text{TOVHSYN}}$ is updated on the rising edge of TOVHCK. If operated in input mode $\overline{\text{TOVHSYN}}$ must be asserted together with the X-bit of the 1st subframe of the DS3 signal which is input on TOVHD. $\overline{\text{TOVHSYN}}$ is sampled on the rising edge of TOVHCK.</p> |
| D9 | TSBCK | O | <p>Transmit Stuff Bit Clock This signal provides the bit clock for DS3 stuff bit data. Transmit stuff bit data is sampled on the rising edge of TSBCK.</p> |
| A7 | TSBD | I | <p>Transmit Stuff Bit Data Data provided via TSBD is optionally inserted in the stuffed bit positions of the DS3 signal. TSBD is sampled on the rising edge of TSBD. This function is available in M13 asynchronous format only.</p> |
| B9 | ROVHCK | O | <p>Receive Overhead Bit Clock This signal provides the bit clock for the received DS3 overhead bits. ROVHCK is nominally a 526 kHz clock.</p> |
| C9 | ROVHD | O | <p>Receive Overhead Data ROVHD contains the extracted overhead bits of the DS3 frame. It is updated on the rising edge of ROVHCK.</p> |
| C10 | $\overline{\text{ROVHSYN}}$ | O | <p>Receive Overhead Synchronization $\overline{\text{ROVHSYN}}$ is asserted while the X-bit of the 1st subframe of the DS3 overhead bits is provided via ROVHD. It is sampled on the rising edge of ROVHCK.</p> |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|--------|-------------------------|--|
| D11 | RSBCK | O | <p>Receive Stuff Bit Clock</p> <p>This signal provides the bit clock for DS3 stuff bit data. Transmit stuff bit data is sampled on the rising edge of TSBCK.</p> |
| A10 | RSBD | O | <p>Receive Stuff Bit Data</p> <p>ROVHD provides data which was inserted in the stuffed bit positions of the DS3 signal. RSBD is updated on the rising edge of RSBD. This function is available in M13 asynchronous format only.</p> |

2.5 Test Interface

| Pin No. | Symbol | Input (I) Output (O) | Function |
|---------|--------------------------|-------------------------|---|
| C25 | TCK | I | JTAG Test Clock This pin is connected with an internal pull-up resistor. |
| F23 | TMS | I | JTAG Test Mode Select This pin is connected with an internal pull-up resistor. |
| A24 | TDI | I | JTAG Test Data Input This pin is connected with an internal pull-up resistor. |
| D24 | TDO | O | JTAG Test Data Output |
| B26 | $\overline{\text{TRST}}$ | I | JTAG Test Reset This pin is connected with an internal pull-down resistor. |
| E24 | SCAN | I | Full Scan Path Test When connected to V_{DD3} the MUNICH256FM works in a vendor specific test mode. It is recommended to connect this pin to V_{SS} . |

2.6 Power Supply, Reserved Pins and No-connect Pins

| Pin No. | Symbol | Input (I) Output (O) | Function |
|--|------------|-------------------------|--|
| AF1, AE7, AF9, AE12, AE15, AF18, AE20, AF26, AD3, AD24, AD26, Y2, Y25, V1, V26, R2, T12, T11, R12, R11, T14, T13, R14, R13, T16, T15, R16, R15, R25, P12, P11, N12, N11, P14, P13, N14, N13, P16, P15, N16, N15, M2, M12, M11, L12, L11, M14, M13, L14, L13, M16, M15, L16, L15, M25, J1, J26, G2, G25, C3, C24, D25, A1, B7, A9, B12, B15, A18, B20, A26, B23, A25 | V_{SS} | I | Ground 0V All pins must have the same level. |
| AE2, AF5, AE10, AF12, AF15, AE17, AF22, AE25, AB1, AB26, Y1, Y26, U2, U25, R1, R26, M1, M26, K2, K25, G1, G26, E1, E26, B2, A5, B10, A12, A15, B17, A22, B25, C22, D21 | V_{DD25} | I | Supply Voltage 2.5V ± 0.25V All pins must have the same level. |
| AC4, AD6, AD9, AC10, AD14, AD18, AC17, AD21, AC23, AA3, AA24, W3, U4, V24, U23, P3, P23, N24, L24, J3, K23, J24, H23, F3, F24, D4, C6, D10, C13, D17, C18, C21, D23 | V_{DD3} | I | Supply Voltage 3.3V ± 0.3V All pins must have the same level. |

Pin Description

| Pin No. | Symbol | Input (I) Output (O) | Function |
|--|------------------------|-------------------------|--|
| B5, C5, D5, A4, B4, C4, E3, D2, H3, H2, J4, H1, J2, K4, K3, K1, F4, D1, E2, G4, F2, G3, F1, H4, L3, L4, L2, L1, M3, M4, N1, N2, AA26, W25, W26, T23, U24, T24, R23, V25, U26, R24, T25, P24, T26, P25, P26, N25, N23, L26, K26, M23, L25, H26, L23, J25, K24, H25, F26, J23, H24, F25, G24, D26, G23, E25, C26, D20, B22, A23, C20, D18, B19, A20, B18, C17, A19, A17, D16, D15, A16, B13, A13, B11, C11, C7, D7, A6, B6, D6 | RES1..16, RES20..93 | | <p>Reserved Pins 1..16, 20..93</p> <p>A pull-up resistor to V_{DD3} is recommended.</p> |
| E4, C1, B1, C2, A3, A2, B3, D3, E23, B24, C23, D22, AC22, AD23, AD22, AC21, AE22, AC20, AF24, AE23, AF2, AE3, AC5, AD4, AE1, AD2, AB4, AC3 | NC0..7 NC12..31 | | <p>No-connect Pins 0..7, 12..31</p> <p>It is recommended not to connect these pins.</p> |

3 General Overview

3.1 Functional Overview

The MUNICH256FM is a highly integrated WAN protocol controller that performs HDLC, PPP and transparent (TMA) protocol processing on 256 full duplex serial channels for a channelized or unchannelized DS3 link. The device provides the framing functions for 28 T1 links or 21 E1 links. Signalling controller functions for DS3, T1 and E1 mode are integrated as well.

The following operating modes are provided (assuming a PCI clock frequency of 33 MHz or more):

- 28 times T1 signals operating at 1.544 MBit/s mapped into M13 asynchronous format or C-bit parity format
- 21 times E1 signals operating at 2.048 MBit/s mapped into ITU-T G.747 compliant signal.
- Full payload rate DS3 signal in C-bit parity format

The serial interface operates in unipolar or dual-rail mode and connects directly to available DS3 LIUs.

Each T1 or E1 tributary can be operated in external timing mode, where the tributary is clocked with the common transmit clock CTCLK, or in looped timing mode, where data of the selected tributaries is sent synchronous to the incoming receive clock.

A variety of loop modes is provided to support remote as well as inloop testing of the device. Remote loops are provided on DS3-, DS2-, DS1- or payload level.

Two bus interfaces, a PCI Rev. 2.1 compliant bus interface and a 16 bit Intel/Motorola style bus interface, connect the device to system environment. Device configuration and channel operation is provided through the PCI bus interface, whereas the 16 bit bus interface provides access to the framing functions and the signalling controller. The MUNICH256FM supports PCI PnP capability by loading the subsystem ID and the subsystem vendor ID via a SPI™ interface into the PCI configuration space.

3.2 Block Diagram

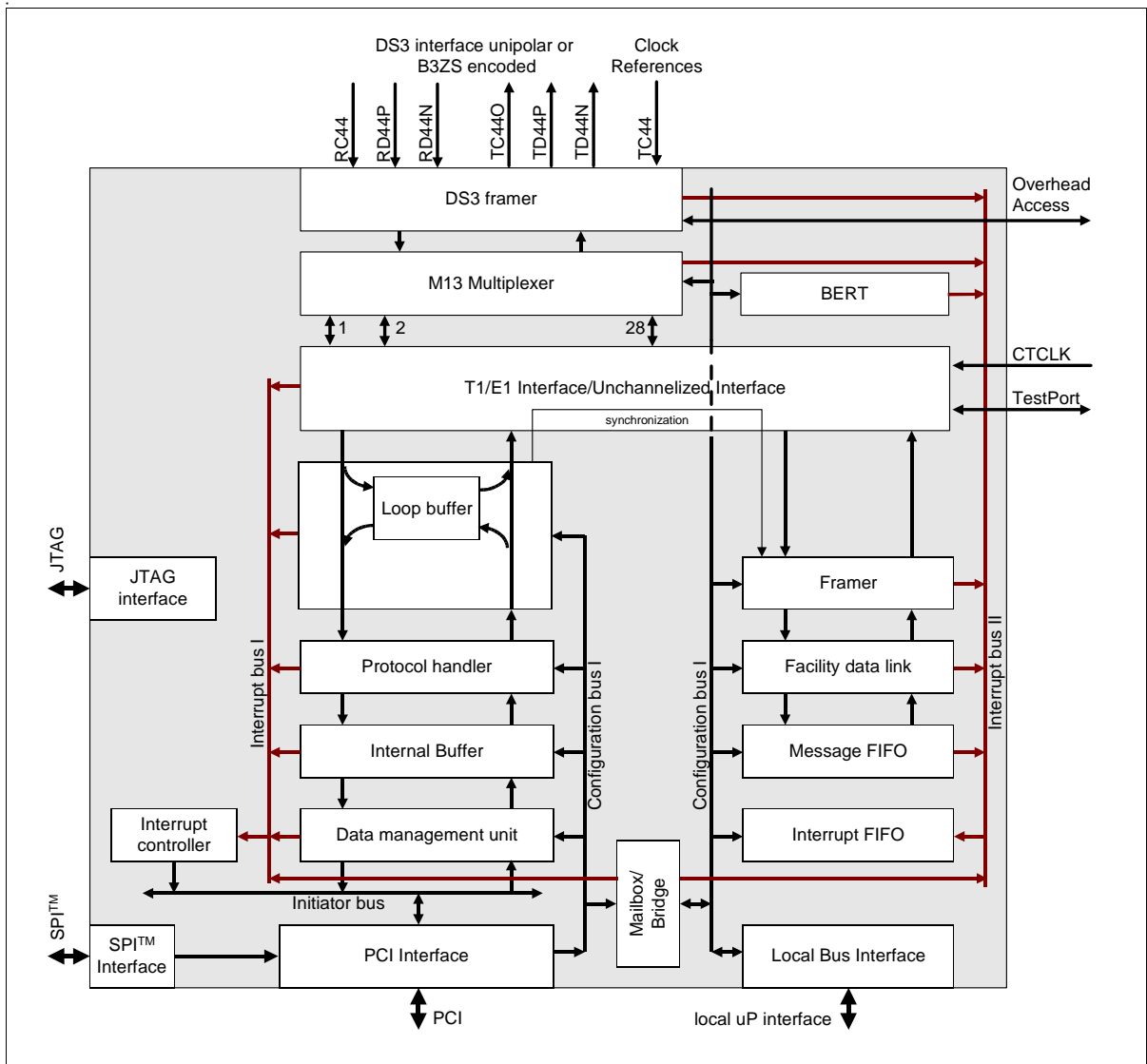


Figure 3-1 MUNICH256FM Block Diagram

3.3 Internal Interface

The device consists of several macro functions as shown in **Figure 3-1**. The internal modules are connected by busses/signals according to Infineon's on-chip bus.

The main busses are:

- The initiator bus, on which the DMA requests of the data management units and the interrupt controller are arbitrated and funneled into the PCI interface.

- The configuration busses, which serve as the standard programming interface to access the chip internal registers and functions either via PCI bus or via the local bus interface.
- The interrupt busses, which collect all interrupt information and forward them to the corresponding interrupt handler.

The chip's core functions are all operated with the PCI clock. Transfers between clocking regions (serial clocks and system clock) are implemented only in the serial interface.

3.4 Block Description

The following section gives a brief overview to the function of each block. For a detailed description of each function refer to "Functional Description" on page 47.

T1/E1 Interface/Unchannelized Interface

The T1/E1 interface consists of the subfunctions receive and transmit. This block provides the function of serial/parallel and parallel/serial conversion for up to 28 incoming and up to 28 tributaries of the DS3 signal. Serial data is transferred between the internal clocking system, which is derived from the PCI clock, and the various line clocks. This provides a unique clocking scheme on the internal interfaces. The aggregate bandwidth of all enabled tributaries can be up to 45 Mbit/s in each direction.

Time slot assigner

The time slot assigner exchanges data with the serial interface on a 8 bit parallel bus, thus funneling all data of up to 28 interfaces. The time slot assigner provides freely programmable mapping of any time slot or any combination of time slots to 256 logical channels. A programmable mask can be provided to allow subchanneling of the available time slots which allows channel data rates starting at 8kbit/s.

At the protocol machine interface the time slot assigner and the protocol machine exchanges channel oriented data (8 bit) together with the time slots masks.

Protocol handler

Two protocol machines, one for receive direction and one for transmit direction, provide protocol handling for up to 256 logical channels and a maximum serial aggregate data rate of up to 45 Mbit/s per direction. The protocol machines implement four modes, which can be programmed independently for each logical channel: HDLC, bit-synchronous PPP, octet-synchronous PPP and Transparent Mode A, including frame synchronous TMA.

Internal buffer

The internal buffers provides channelwise buffering of raw (unformatted/deformatted) data for 256 logical channels. Channel specific thresholds can be programmed

independently in transmit and receive direction. In order to avoid transmit underrun conditions each transmit channel has two control parameters for smoothing the filling/emptying process (transmit forward threshold, transmit refill threshold). In receive direction each channel has a receive burst threshold. To avoid unnecessary waste of bus bandwidth, e.g. in case of transmission errors, the receive buffer provides the capability to discard frames which are smaller than a programmable threshold.

Data management units

The data management units provide direct data transfer between the system memory and the internal buffers. Each channel has an associated linked list of descriptors, which is located in system memory and handled by the data management units. This linked list is the interface between the system processor and the MUNICH256FM for exchange of data packets. The descriptors and the data packets can be stored arbitrarily in 32 bit address space of system memory, thus allowing full scatter/gather assembly of packets. In order to optimize PCI bus utilization, each descriptor is read in one burst and hold on-chip afterwards.

Interrupt controller

Two interrupt controller manage internal interrupts. Interrupts from the mailbox, the framing engines and the signalling controller are passed in form of interrupt vectors to an internal interrupt FIFO which can be read from the local bus. All system, port and channel related interrupt informations are passed to the main interrupt controller which is connected to the PCI system. A programmable DMA with nine channels stores these interrupts in form of interrupt vectors in different interrupt queues in system memory.

PCI interface

The PCI interface unit combines all DMA requests from the internal data management unit and the interrupt controller and translates them into PCI Rev. 2.1 compliant bus accesses. The PCI interface optionally includes the function of loading the subsystem vendor ID and the subsystem ID from an external SPI compliant EEPROM.

Mailbox, internal bridge and global registers

The mailbox is used to exchange data between the PCI attached microprocessor and the local bus microprocessor and provides a doorbell function between the two interfaces.

Controlled by an arbiter an internal bridge connects the configuration bus I and the configuration bus II. It is therefore possible to access the "layer one" registers from the PCI interface directly. Thus the device could also be operated without a local microprocessor connected to it, e.g. for debugging purposes. It is NOT possible to access the configuration bus I and therefore the 'HDLC' registers or the PCI bridge from the local bus.

Local bus interface

The local bus interface builds the interface between the local microprocessor and the on-chip configuration bus II in order to access the registers of the on-chip M13 multiplexer, DS2/DS3 framer, T1/E1 framer, the registers of the signalling controller and the mailbox. The local bus interface provides a switchable Intel-style or Motorola-style processor interface.

M23 multiplexer/demultiplexer and DS3 framer

In channelized operating modes the M23 multiplexer/demultiplexer maps/demaps seven DS2 signals into/from M13 asynchronous format or C-bit parity format. In unchannelized mode one logical input stream is mapped into the information bits of the DS3 stream according to ANSI T1.107. The DS3 framer performs frame and multiframe alignment in receive direction and inserts the frame and multiframe alignment bits. Performance monitors provide for counting of framing bit errors, parity errors, CP-bit errors, far end block errors, excessive zeroes or line code violations. The framer detects loopback requests and allows insertion of loopback requests under microprocessor control.

M12 multiplexer/demultiplexer and DS2 framer

The M12 multiplexer/demultiplexer operates in two modes. It maps either 28 T1 signals or 21 E1 signals into/from seven ANSI T1.107 or ITU-T G.747 compliant DS2 signals. It performs inversion of the second and fourth DS1 signal. The DS2 framer performs frame and multiframe alignment in receive direction and vice versa inserts the framing bits according to ANSI T1.107 or ITU-T G.704. It detects loopback requests or enables insertion of loopback requests under microprocessor control.

T1/E1 framer

Synchronization is achieved with the on-chip framing function. T1/E1 mode is supported for up to 28 ports. Once the framer achieved synchronization for a line, that is the frame alignment information in the incoming bit stream has been identified correctly, it informs the port interface and the facility data link about the frame position. In transmit direction the framing bits are inserted according to T1 F4 format, T1 SF (F12) format, T1 ESF (F24) format, E1 doubleframe format or E1 CRC-4 multiframe format. Performance monitors provide for counting framing errors, CRC errors, block errors, E-bit errors or PRBS bit errors. The framer detects loopback requests and allows insertion of loopback requests or pseudo-random bit sequences under microprocessor control.

Facility data link, Signaling controller

The facility data link exchanges the 'F-bits' of the T1 links or the S_a -bits of time slot zero of the E1 links with the framer block and it provides the function of HDLC formatting or BOM mode in receive and transmit direction.

The signalling controller also provides access to the DS3 signalling bits (Far End Alarm and Control Channel, Path Maintenance Data Link Channel).

Message FIFO

For intermediate buffering of data link messages two FIFOs are integrated, one for transmit and one for receive direction. Each FIFO provides two pages of 32 bytes buffer per line and direction.

JTAG

Boundary Scan logic according to IEEE 1149.1.

4 Functional Description

4.1 Port Handler

The port handler is the interface between the serial ports and the chip internal protocol and framing functions. It converts incoming serial data into parallel data for further internal processing and in the outgoing direction it converts parallel data into a serial bit stream.

The MUNICH256FM provides one port for operation at DS3 signal speeds. It provides unipolar data transmission or B3ZS encoded data transmission.

The system interface consists of one receive clock input and either one receive data input in unipolar mode or two receive data inputs in dual-rail mode, one for the positive pulse and one for the negative pulse. In transmit direction the system interface is build of one transmit clock input and one or two transmit data outputs.

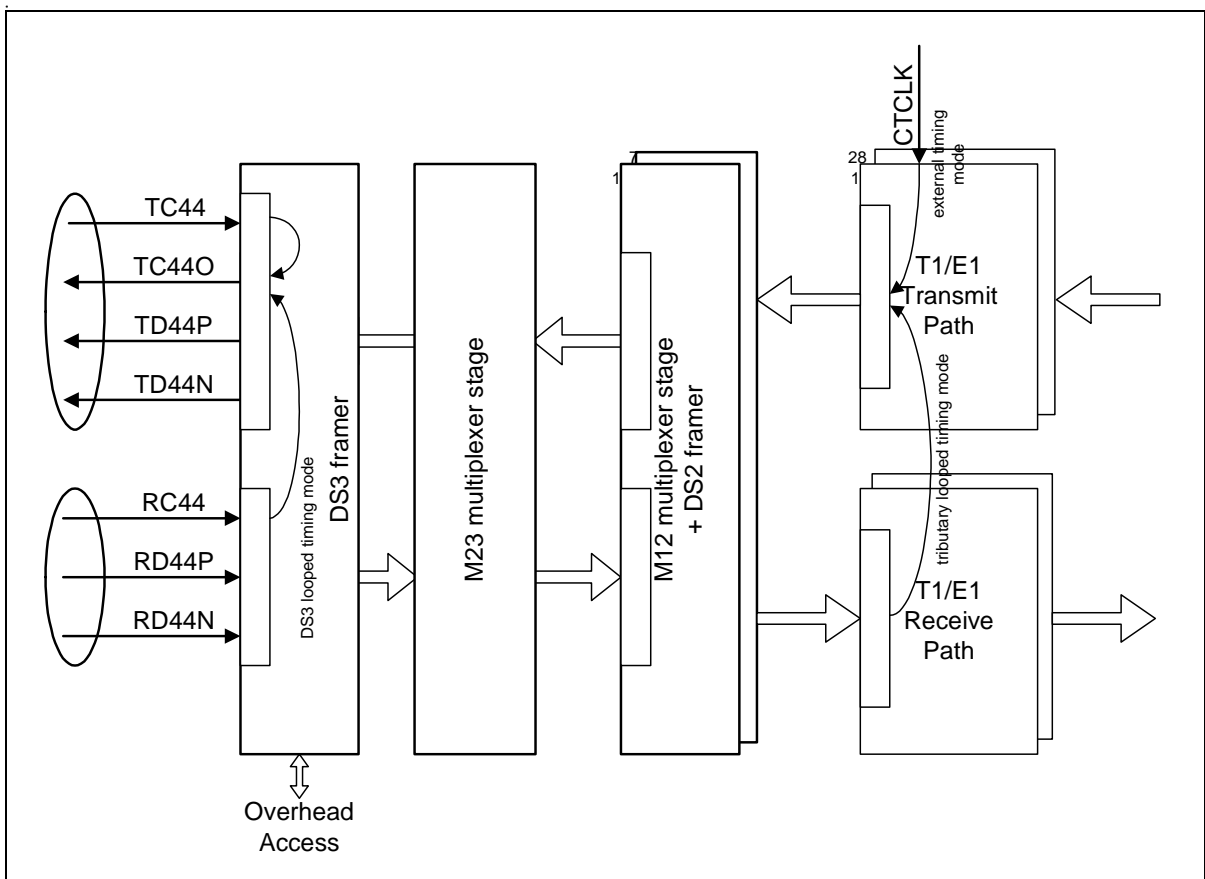


Figure 4-1 Port configuration in M13 mode

4.1.1 Local Port Loop

Local port loops are provided on DS3, DS2 and DS1 level on a per port/tributary basis. In the local loop the outgoing bit stream of a port/tributary is mirrored to the receive data path. This allows to prepare data in system memory, which is processed by the MUNICH256FM in transmit direction, mirrored to the respective receiver and stored in system memory again. In order to ensure that the local port loop works even without incoming receive clock, each receiver looped uses the corresponding transmit clock.

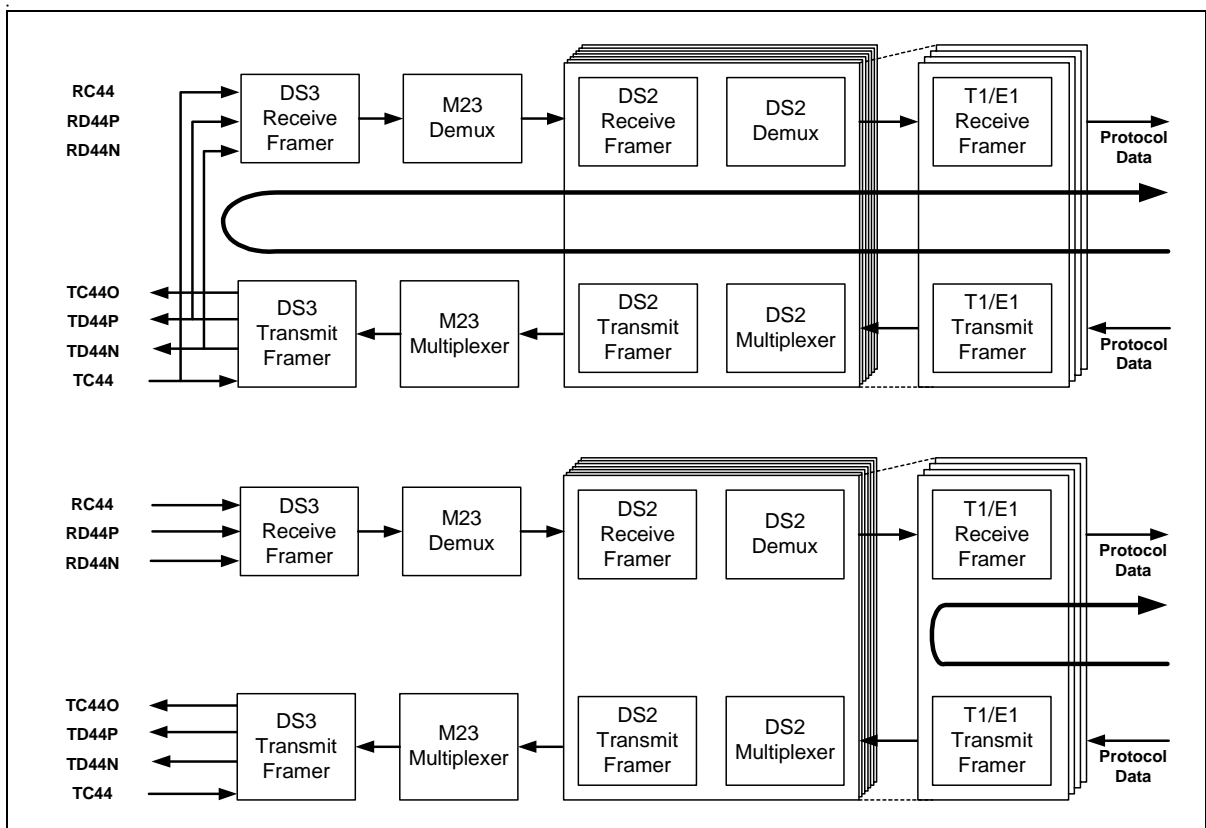


Figure 4-2 Local Port Loops in M13 mode

4.1.2 Remote Line Loops

The MUNICH256FM supports remote line loops in different stages of the M13 data path. In DS3 line loopback mode the incoming DS3 signal is mirrored and placed on the DS3 signal output. While operating in DS3 line loopback mode, the incoming receive clock RCLK is used to update outgoing transmit data. In DS2 line loopback mode one arbitrarily selectable DS2 signals is looped in the M12 stage of the MUNICH256FM. The T1/E1 line loopback mode mirrors one or more incoming lines. Transmit data coming from the transmit data path is replaced with the mirrored data stream.

Functional Description

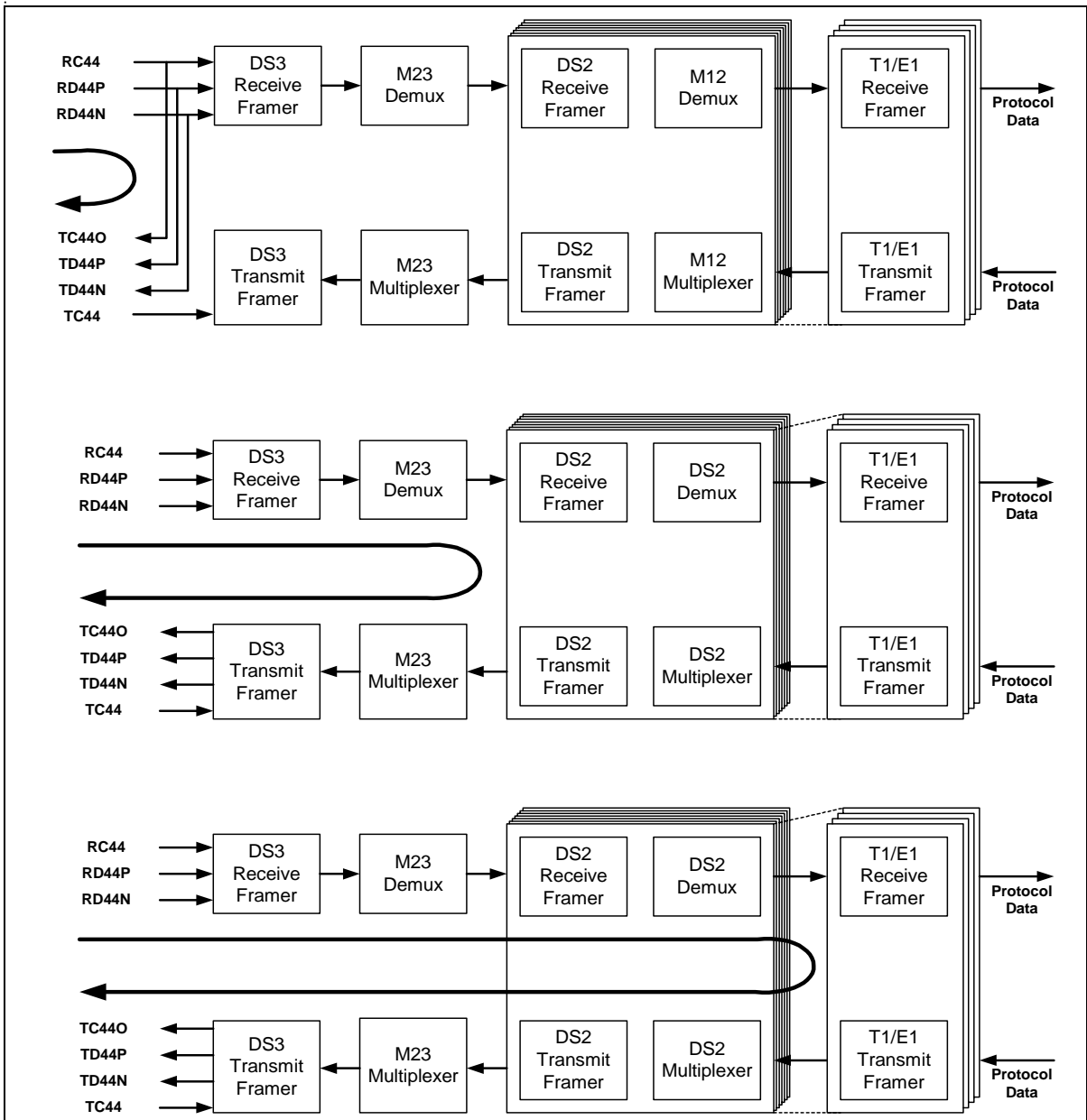


Figure 4-3 Remote Line Loops

The T1/E1 line loopback mode mirrors one or more incoming lines. Transmit data coming from the transmit data path is replaced with the mirrored data stream. While T1/E1 line loop is closed the transmit framer and the protocol machines are disabled.

4.1.3 Test Breakout

The test breakout function provides the capability to multiplex one of the incoming 28 receive tributaries to the outgoing test receive port, where an external T1/E1 analyzer can be easily connected to. A selectable incoming tributary signal can be mapped to the test receive port where RCLK(x) is mapped to TRCLK and RD(x) to TRD. TRD is updated on the falling edge of TRCLK. In the opposite direction one of the 28 transmit tributaries can be replaced with the incoming test transmit data input TTD and the test transmit clock input TTCLK. TTD is sampled on the rising edge of TTCLK.

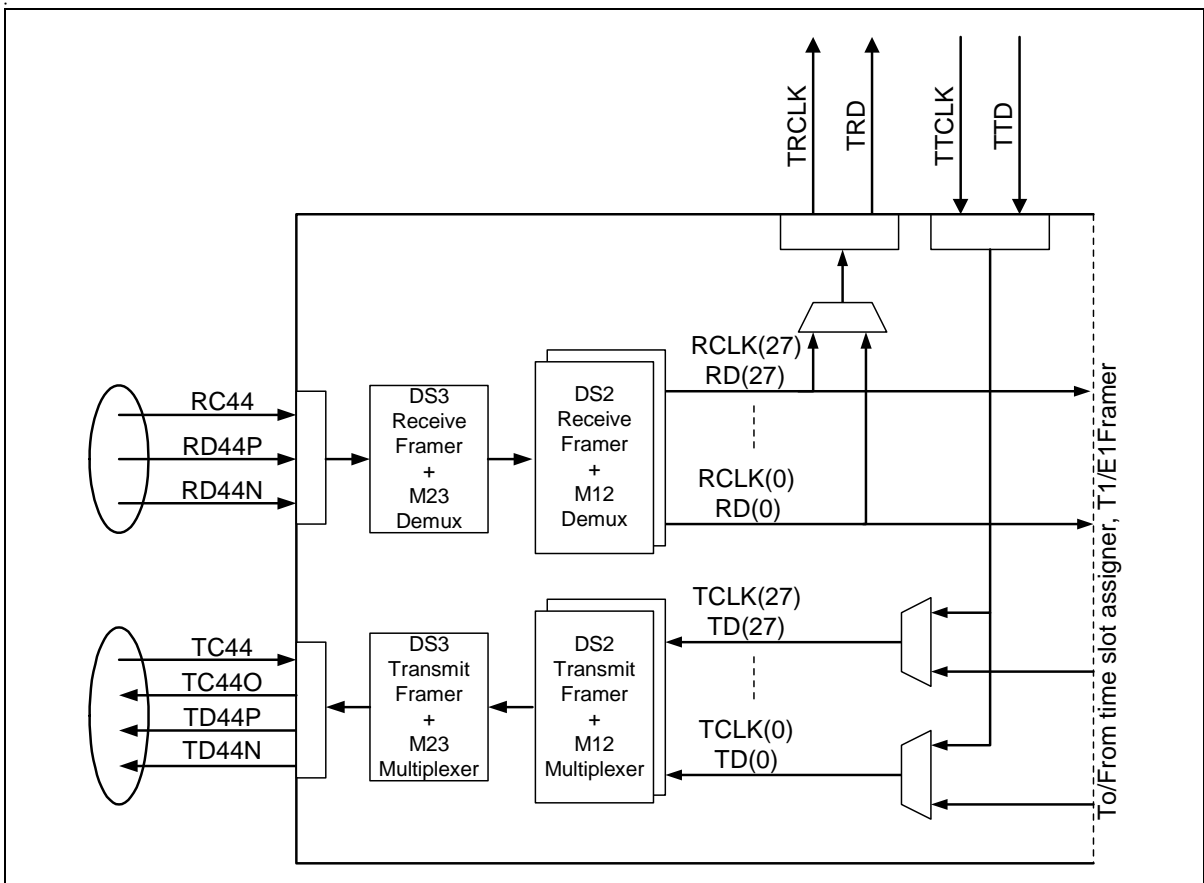


Figure 4-4 Test Breakout

4.2 Time slot Handler

4.2.1 Channelized Modes

The time slot handler assigns any combination of time slots of ports configured in T1 or E1 mode to logical channels. The assigned time slots are connected internally and the bit stream of one logical channel is mapped continuously over the selected time slots. Since the receiver and the transmitter operate independently of each other, the assignment of time slots to logical channels can be done separately in receive and transmit direction. Any time slot can be assigned to any channel and any sequence of time slots can be assigned to one channel.

In normal operation each time slot consists of eight bits and all bits are used for data transmission. An available mask function provides the capability to mask selected bits, which in turn are disabled for data transmission. This provides the possibility to operate time slots with less than 64 kBit/s throughput. So, instead of mapping the bit stream of one logical channel over all bits of the assigned time slots, the bit stream is mapped continuously over all unmasked bits of the time slots belonging to that channel.

Figure 4-5 shows a simple assignment process. In this case one port is configured in E1 mode and time slots two and three are assigned to logical channel 5. The bit mask of time slot two is set to FE_H , which disables bit zero of that time slot, and the bit mask of the third time slot is set to FD_H , which disables bit one.

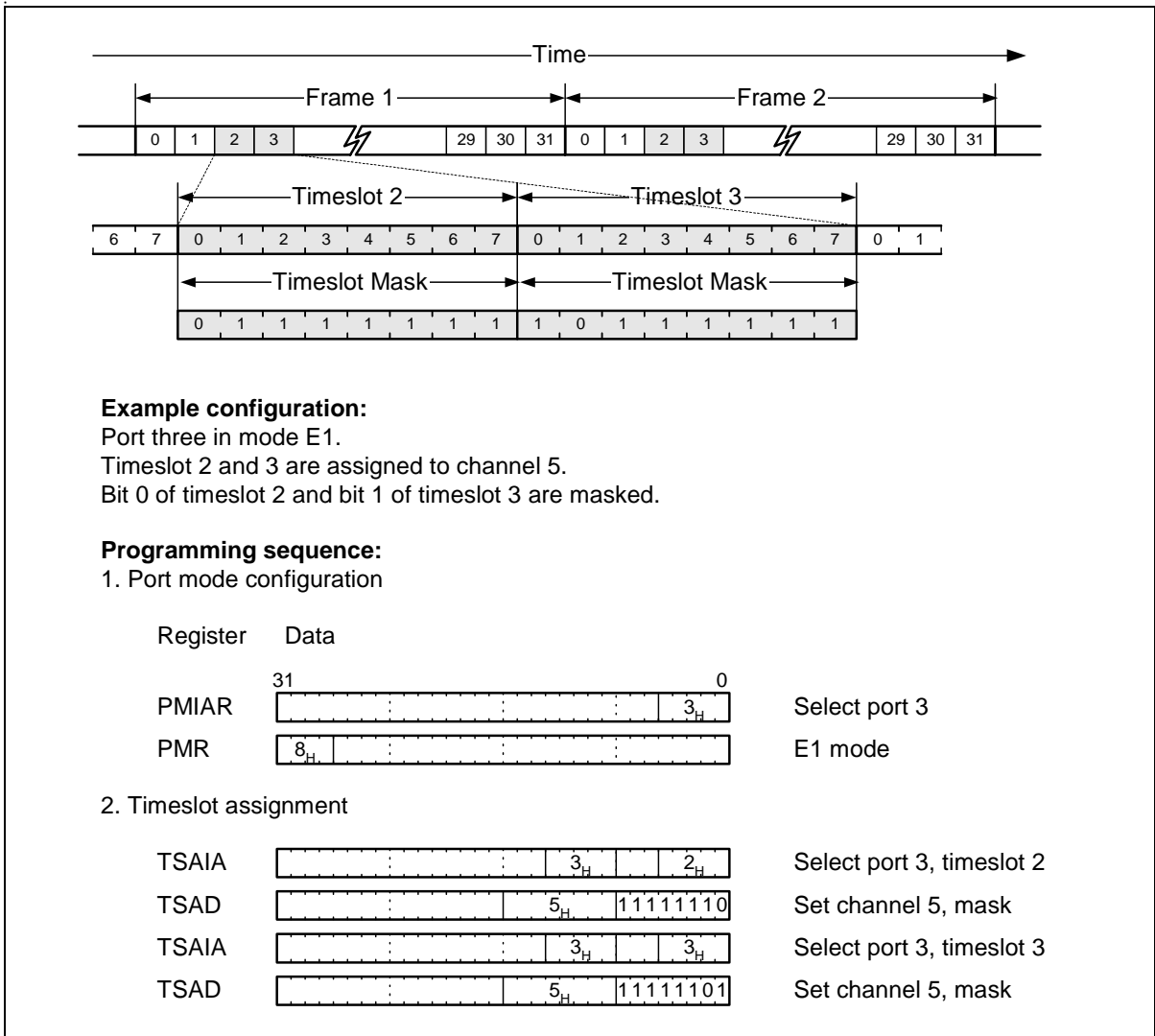


Figure 4-5 Time slot Assignment in Channelized Modes

4.2.2 Unchannelized Mode

In unchannelized mode the complete incoming and outgoing serial bit stream belongs to one logical DS3 channel. To operate the link in unchannelized mode tributary zero (port zero) has to be programmed for unchannelized operation and all 'time slots', that is time slot 0 to 23 must be assigned to one channel. Additionally the M13 multiplexer must be switched into unchannelized DS3 mode. The function of bit masks, which is available for the T1/E1 tributaries, is not available in unchannelized mode.

4.3 Data Management Unit

Each packet or part of a packet is referenced by a descriptor. The descriptors form a link list, thus connecting all packets together. Packet data as well as descriptors are located in system memory. Both the MUNICH256FM and the system CPU operate on these data structures.

Each logical channel has its dedicated linked list of descriptors, one for receive direction and one for transmit direction. This type of data structure allows channel specific memory organization which can be specified by the system processor. It provides an optimized way to transfer data packets between the system processor and the MUNICH256FM.

The MUNICH256FM has a flexible DMA controller to transfer data either from the internal receive buffer to the shared memory (receive direction) or from the shared memory to the internal transmit buffer (transmit direction). Each DMA works on one linked list. Each linked list located in system memory is associated with one of the 256 transmit channels or one of 256 receive channels.

The address generator of the DMA controller supports full link list handling. Descriptors are stored independently from the data buffers, thus allowing full scatter/gather assembly and disassembly of data packets.

4.3.1 Descriptor Concept

A descriptor is used to build a linked list, where each member of the linked list points to a data section. A descriptor consists of four DWORDS¹⁾. The first three DWORDS, containing link and packet information, are provided by the system CPU and the last DWORD contains status information, which is written when the MUNICH256FM has finished operation on a descriptor.

The data section itself can be of any size up to the maximum size of 65535 bytes per descriptor and is defined in the first DWORD of a descriptor. Each logical data packet can be split into one or multiple parts, where each part is referenced by one descriptor, and all parts are referenced by a linked list of descriptors. The descriptor containing the last part of a data packet is marked with a frame end bit. The descriptor following the marked descriptor therefore contains the beginning of the next data packet (**Figure 4-6**). The last descriptor in a linked list is marked with a hold indication.

For ease of programming the transmit descriptor and the receive descriptor are structured the same way, thus allowing to link a receive descriptor directly into the linked list of the transmit queues with minimum descriptor processing.

1)

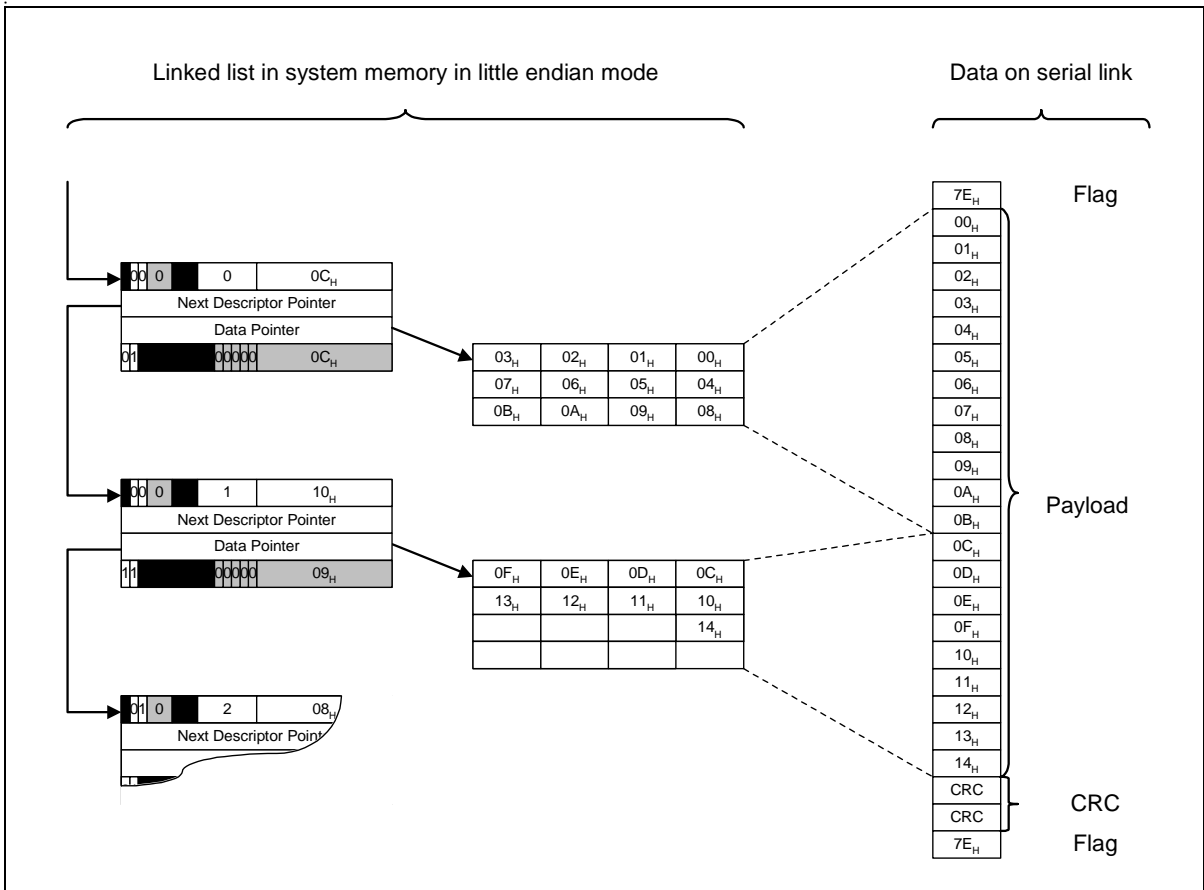


Figure 4-6 Descriptor Structure

Although the data management unit works 32-bit oriented, it is possible to begin a transmit data section at an uneven address. The two least significant bits of the transmit data pointer determine the beginning of the data section and the number of bytes in the first DWORD of the data section, respectively. In receive direction the address of the data sections must be DWORD aligned.

4.3.2 Receive Descriptor

Each receive descriptor is initialized by the host CPU and stored in system memory as part of a linked list. The MUNICH256FM reads a descriptor, when requested so from the host by a receive command or after branching from one receive descriptor to the next receive descriptor. Each receive descriptor contains four DWORDs, where the first three DWORDs contain link and packet information and the last DWORD contains status information. Once the descriptor is processed the status information will be written back to system memory by the MUNICH256FM (Receive status update). When the

Functional Description

MUNICH256FM branches to a new descriptor it reads the link and packet information entirely and stores it in its on-chip channel database.

Table 4-1 Receive Descriptor Structure

| DWORD ADDR. | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------------|------------------------------------|------|-----|-------------|----|----|----|----|----|----|-------------------|-----|------|-----|------|-----|
| 00 _H | 0 | HOLD | RHI | OFFSET(2:0) | | | 0 | 0 | 0 | 0 | DescriptorID(5:0) | | | | | |
| 04 _H | NextReceiveDescriptorPointer(31:2) | | | | | | | | | | | | | | | |
| 08 _H | ReceiveDataPointer(31:2) | | | | | | | | | | | | | | | |
| 0C _H | FE | C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MFL | RFOD | CRC | ILEN | RAB |

| DWORD ADDR. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------------|------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 00 _H | NO(15:0) | | | | | | | | | | | | | | | |
| 04 _H | NextReceiveDescriptorPointer(31:2) | | | | | | | | | | | | | | 0 | 0 |
| 08 _H | ReceiveDataPointer(31:2) | | | | | | | | | | | | | | 0 | 0 |
| 0C _H | BNO(15:0) | | | | | | | | | | | | | | | |

HOLD

Hold indication

HOLD indicates that a descriptor is the last element of a linked list containing valid information.

- 0 Next descriptor is available in the shared memory. After checking the HOLD bit the data management unit branches to the next receive descriptor.
- 1 This descriptor is the last one that is available for a channel. This means that the data section where this descriptor points to is the last data section which is available for data storage. After processing of descriptor has finished, the data management unit repolls the descriptor one time to check if HOLD has already been cleared. If HOLD is still set the corresponding receive channel is deactivated as long as the system CPU does not request a new activation via a 'Receive Hold Reset' command or forces the MUNICH256FM to branch to a new linked list via a 'Receive Abort/Branch' command.

Note: When repolling a descriptor the MUNICH256FM checks the HOLD bit and the bit field NextReceiveDescriptorPointer. All other information are NOT updated in the internal channel database.

Functional Description

| | |
|--------------|--|
| RHI | <p>Receive Host Initiated Interrupt</p> <p>This bit indicates that the MUNICH256FM shall generate a 'Receive Host Initiated' interrupt vector after it has finished processing the descriptor.</p> <p>0 Data management unit does not generate an interrupt vector after it has processed the receive descriptor.</p> <p>1 Data management unit generates an interrupt vector, as soon as all data bytes are transferred into the current data section and the status information is updated.</p> |
| OFFSET | <p>Offset of unused data section.</p> <p>This bit field allows to reserve memory space in increments of DWORDs for an additional header. If the marked descriptor is the first one of a new packet the data management unit will write data at the address <code>ReceiveDataPointer+4xOFFSET</code>.</p> <p><i>Note: Offset x 4 must be smaller than NO.</i></p> <p><i>Note: This option is not available in transparent mode.</i></p> |
| DescriptorID | <p>This bit field is read by the data management unit and written back in the corresponding interrupt status of a channel interrupt vector which is generated by the data management unit. This value provides a link between the descriptor and the corresponding interrupt vector.</p> |
| NO | <p>Byte Number</p> <p>This bit field defines the size of the receive data section allocated by the host. The maximum buffer length is 65535 bytes and it has to be a multiple of 4 bytes. Data bytes are stored in the receive data section according to the selected mode (little endian or big endian).</p> <p><i>Note: Please note that the device handles the status (CRC, flag and frame status) of frame based protocols (HDLC, PPP) internally in the same way as payload data. Therefore byte number should include four bytes more than the maximum length of incoming frames. Nevertheless, the frame status will be deleted from the end of the data stream and be attached as a status word to the receive descriptor. The frame status will not be written to the data section.</i></p> |

NextReceiveDescriptorPointer

This pointer contains the start address of the next valid receive descriptor. After completion of the current receive descriptor the data management unit branches to the next receive descriptor to continue data reception.

System CPU can force the MUNICH256FM to branch to the beginning of a new linked list via the command 'Receive Abort/Branch'. In this case the receive descriptor address provided via register CSPEC_FRDA is used as the next receive descriptor pointer to be branched to.

ReceiveDataPointer

This pointer contains the start address of the receive data section. The start address must be DWORD aligned.

FE Frame End

It indicates that the current receive data section (addressed by ReceiveDataPointer) contains the end of a frame. This bit is set by the data management unit after transferring the last data of a frame from the internal receive buffer into the receive data section which is located in the shared memory. Moreover the bit field BNO and the status bits are updated, the complete (C) bit is set and a 'Frame End' interrupt vector is generated.

C Complete

This bit indicates that

- filling the data section has completed (with or without errors),
- processing of this descriptor was aborted by a 'Receive Abort/Branch' command,
- or the end of frame (PPP, HDLC) was stored in the receive data section.

The complete bit releases the descriptor.

BNO Byte Number of Received Data

The data management unit writes the number of data bytes stored in the current data section into bit field BNO.

Functional Description

When the MUNICH256FM completes a data section, which included the end of a frame (C bit and FE bit are set), or when the MUNICH256FM branches to a new linked list due to a 'Receive Abort/Branch' command the status information bits RAB, ILEN, CRC, RFOD and MFL are updated as part of the receive status update. In the abort scenario, the C bit will always be set. Bit FE will be set only, if the particular channel operates in HDLC or PPP mode.

| | |
|------|--|
| RAB | Receive Abort This bit is set when <ul style="list-style-type: none"> •the incoming serial data stream contained an abort sequence, or •an incoming frame was aborted by the command 'Receive Abort/Branch', or •when a channel is switched off while a frame is being received. |
| ILEN | Illegal length This bit is set, when the length of the incoming data packet was not a multiple of eight bits. |
| CRC | CRC Error This bit is set, when the checksum of an incoming data packet was different to the internally calculated checksum. |
| RFOD | Receive Frame Overflow This bit is set, when a receive buffer overflow occurred during data reception. |
| MFL | Maximum Frame Length This bit is set, when the length of the incoming data packet exceeded the value programmed in CONF1.MFL. |

4.3.3 Data Management Unit Receive

The *data management unit receive* transfers data for each of the 256 logical receive channels from the internal receive buffer to the data sections of the corresponding channel. To fulfill the task it has to be initialized for operation, which is described in "Channel Programming / Reprogramming Concept" on page 162. Relevant part of the channel information for the data management unit is the address pointer to the first receive descriptor, the channel interrupt queue and the channel interrupt mask.

The first receive descriptor of a channel is fetched from system memory and stored in the chip internal channel database the first time the receive buffer requests a data transfer for the channel. The descriptor contains a pointer to the data section, the size of the provided data section and a pointer to the next receive descriptor.

The data transfer is requested as soon as a programmed receive buffer threshold is reached. This threshold is programmed during channel setup on a per channel basis. Task of the data management unit is to calculate the maximum number of bytes that can

Functional Description

be stored in the receive data section and to compare this with the length of the requested data transfer.

In case that the requested transfer length from the receive buffer fits into the provided data section the data management unit transfers the data block to system memory in one single burst. If the requested transfer length exceeds the available space of the data section the transfer is divided into two or more parts. Data packets are written to the data section until the given data section is filled or the end of a packet is reached.

If the data section in the shared memory is completely filled with data, the data management unit updates the status word of the receive descriptor by setting the complete (C) bit and the number of bytes (BNO), which are stored in the data section. In this case the number of bytes written to the data section equals the size of the data section.

If the data packet, which is written to system memory, contains the remaining part of a completely received packet, the data management unit updates the status word of the receive descriptor by setting the complete bit together with the frame end (FE) bit. The BNO field is updated on the actual value of bytes written to the data section. If enabled, the data management unit generates a 'Frame End' channel interrupt vector.

With the next receive buffer request the data management unit branches to the next receive descriptor, which was referenced in the next descriptor field of the current processed descriptor. To keep track of the linked list the data management unit provides the possibility to issue a 'Receive Host Initiated' interrupt vector, which is generated after the status word was updated. To enable this interrupt vector the bit RHI must be set in a descriptor.

Descriptor hold operation

Processing of the descriptor list is controlled by the HOLD bit, which is located in the first DWORD of each receive descriptor. The HOLD bit indicates that the marked descriptor is the last descriptor containing a valid data buffer. The data management unit will not branch to a next descriptor until the hold condition is removed or a 'Receive Abort' command forces the MUNICH256FM to branch to the beginning of a new linked list. Since the HOLD bit marks the last descriptor in a linked list, it may prevent that further received data packets can be written to system memory.

When a given data section is filled, does not contain the end of a frame (frame based protocols) and the requested transfer length could not be satisfied, the data management unit polls the HOLD bit of the current receive descriptor once more. If the HOLD bit is removed, it branches to the next descriptor. When the HOLD bit is still '1', an internal poll bit is set and the data management unit does not branch to the next descriptor. Additionally a 'Hold Caused Receive Abort' interrupt vector is generated. The status of the descriptor in the shared memory is aborted (RAB bit set) and the complete bit and the frame end bit are set in the receive descriptor. The rest of the frame will be discarded. As long as the HOLD bit remains set further data of the same channel is

Functional Description

discarded and for each discarded frame a 'Silent Discard' interrupt vector with the bits HRAB and RAB set is generated.

If the current data section was filled and does contain the end of frame a 'Frame End' interrupt vector is generated and the descriptor is updated on the FE bit and the C bit. Therefore the status of this receive descriptor is error free. With the next request of the receive buffer, the data management unit repolls the HOLD bit of the current receive descriptor. If the hold bit is removed, it branches to the next descriptor. If the HOLD bit is still '1', an internal poll bit is set. As long as the HOLD bit remains set, further data of the same channel is discarded and for each discarded frame a 'Silent Discard' interrupt vector with bits HRAB and RAB set is generated.

When the receive buffer request matches exactly the remaining size of the data section and the data block does not contain the end of a packet, it is stored completely in the data section. The descriptor is updated immediately (C bit set). With the next receive buffer request, the data management unit repolls the HOLD bit of the current receive descriptor. If the HOLD bit is removed, it branches to the next descriptor. If the HOLD Bit is still '1', an internal poll bit is set. Additionally a 'Hold Caused Receive Abort' interrupt vector is generated and the rest of the frame is discarded. As long as the HOLD bit remains set further data of the same channel is discarded and for each discarded frame a 'Silent Discard' interrupt vector is generated.

The system CPU can remove the hold condition, when the next receive descriptor is available in shared memory. Therefore the CPU has to execute a 'Receive Hold Reset' command, which will reactivate the channel. When the receive buffer requests a new data transfer, the data management unit will repoll the last receive descriptor. If the HOLD bit was removed, the data management unit branches to the next receive descriptor pointed to by bit field NextReceiveDescriptor.

Note: In protocol modes HDLC and PPP data from receive buffer is discarded until the end of a received frame is reached. As soon as the beginning of a new frame is received, the data management unit starts to fill the data section.

Note: In transparent mode data transferred from receive buffer is written immediately to the data section of the next receive descriptor.

If the CPU issues a 'Receive Hold Reset' command and does not remove the HOLD bit (erroneous programming), no action will take place.

4.3.4 Transmit Descriptor

The transmit descriptor in shared memory is initialized by the host CPU and is read afterwards by the MUNICH256FM. The address pointer to the first transmit descriptor is stored in the on-chip channel database, when requested to do so by the host CPU via the 'Transmit Init' command. The first three DWORDs of a transmit descriptor are read when the transmit buffer requests a data transfer for this channel and then they are stored in the on-chip memory. Also they are read when branching from one transmit

Functional Description

descriptor to the next transmit descriptor. Therefore all information in the next descriptor must be valid when the data management unit branches to a descriptor. The last DWORD of a transmit descriptor optionally is written by the MUNICH256FM when processing of a descriptor has finished.

Table 4-2 Transmit Descriptor Structure

| DWORD ADDR. | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|-----------------|-------------------------------------|------|-----|-----|----|----|----|----|----|----|-------------------|----|----|----|----|----|
| 00 _H | FE | HOLD | THI | CEN | 0 | 0 | 0 | 0 | 0 | 0 | DescriptorID(5:0) | | | | | |
| 04 _H | NextTransmitDescriptorPointer(31:2) | | | | | | | | | | | | | | | |
| 08 _H | TransmitDataPointer(31:0) | | | | | | | | | | | | | | | |
| 0C _H | 0 | C | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

| DWORD ADDR. | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|-----------------|-------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 00 _H | NO(15:0) | | | | | | | | | | | | | | | |
| 04 _H | NextTransmitDescriptorPointer(31:2) | | | | | | | | | | | | | | 0 | 0 |
| 08 _H | TransmitDataPointer(31:0) | | | | | | | | | | | | | | | |
| 0C _H | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

FE Frame end
It indicates that the current transmit data section (addressed by transmit data pointer) contains the end of a frame. After the last byte is read from system memory this bit is passed to the transmit buffer and to the protocol machine. The bit FE informs the transmit buffer to move a stored frame to the protocol machine even if the programmed transmit forward threshold is not reached (see "Internal Transmit Buffer" on page 68). The protocol machine is informed to append the checksum (HDLC, PPP) and then to send the interframe time-fill. Providing a transmit descriptor with FE = '0' and HOLD = '1' is an error.

HOLD Hold indication
It indicates that this descriptor is the last valid element of a linked list.

- 0 Next descriptor is available in the shared memory. The data management unit branches to the next descriptor as soon as processing of the current descriptor has finished.
- 1 The current descriptor is the last descriptor containing valid data in the data section. As soon as the data management unit has transferred the data contained in the data section to the internal buffer, it tries one more time to read the descriptor. In case that

Functional Description

the hold indication is still set, it stores further requests of the receive buffer in its channel database. The channel can be reactivated by issuing a 'Transmit Hold Reset' command or by providing a new linked list via the 'Transmit Abort/Branch' command, in which case not served requests are processed.

Note: When repolling a descriptor the MUNICH256FM checks the HOLD bit and the bit field NextTransmitDescriptorPointer. All other information are NOT updated in the internal channel database.

| | | | | | |
|-------------------------------|--|---|--|---|--|
| NO | <p>Byte Number</p> <p>The byte number defines the number of bytes stored in the data section to be transmitted. Thus the maximum length of data buffer is 65535 bytes. In order to provide dummy transmit descriptors NO = 0 is allowed in conjunction with the FE bit set. In this case (NO = 0) a 'Transmit Host Initiated' interrupt vector and/or the C-bit will be generated/set when the data management unit recognizes this condition. It is an error to set NO = 0 without FE bit set.</p> | | | | |
| THI | <p>Transmit Host Initiated Interrupt</p> <p>This bit indicates that the MUNICH256FM shall generate a 'Transmit Host Initiated' interrupt vector after it has finished operating on the descriptor.</p> <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">0</td> <td>Data management unit does not generate an interrupt vector after it has processed the transmit descriptor.</td> </tr> <tr> <td>1</td> <td>Data management unit generates an interrupt vector, as soon as all data bytes are transferred to the internal transmit buffer and the status information is updated.</td> </tr> </table> | 0 | Data management unit does not generate an interrupt vector after it has processed the transmit descriptor. | 1 | Data management unit generates an interrupt vector, as soon as all data bytes are transferred to the internal transmit buffer and the status information is updated. |
| 0 | Data management unit does not generate an interrupt vector after it has processed the transmit descriptor. | | | | |
| 1 | Data management unit generates an interrupt vector, as soon as all data bytes are transferred to the internal transmit buffer and the status information is updated. | | | | |
| DescriptorID | <p>This bit field is read by the data management unit and written back in the corresponding interrupt status of a channel interrupt vector which is generated by data management unit. This value provides a link between the descriptor and the corresponding interrupt vector.</p> | | | | |
| NextTransmitDescriptorPointer | <p>This pointer contains the start address of the next transmit descriptor. It has to be DWORD aligned. After sending the indicated number of data bytes, the data management unit branches to the next transmit descriptor. The transmit descriptor is read entirely at the beginning of transmission and stored in on-chip memory. Therefore all informations in the descriptor must be valid.</p> <p>System CPU can force the MUNICH256FM to branch to the beginning of a new linked list via the command 'Transmit Abort/Branch'. In this case the transmit descriptor address provided via register</p> | | | | |

Functional Description

CSPEC_FTDA is used as the next transmit descriptor pointer to be branched to.

TransmitDataPointer

This 32-bit pointer contains the start address of the transmit data section. Although the data management unit works DWORD oriented, it is possible to begin transmit data section at byte addresses.

CEN

Complete Enable

This bit is set by the CPU if the complete bit mechanism is desired:

0 The data management unit will NOT update the transmit descriptor with the C bit. In this mode the use of the TH1 interrupt is recommended.

1 The data management unit will set the C bit.

C

Complete

This bit is set by the data management unit, when the bit CEN of a descriptor is set and when it

- completed reading a data section normally, or
- it was aborted by a 'Transmit Off' command or by a 'Transmit Abort/Branch' command.

The complete bit releases the descriptor.

4.3.5 Data Management Unit Transmit

The *data management unit transmit* provides the interface between system memory on one side and the internal transmit buffer on the other side. The data management unit handles requests of the transmit buffer, controls the address and burst length calculation, initiates data transfers from system memory to the transmit buffer and handles the linked lists on a per channel basis.

For initialization the CPU programs the first transmit descriptor address, the interrupt mask, the interrupt queue and starts the channel with the 'Transmit Init' command. For detailed description of channel commands refer to "Channel Commands" on page 163. The data management unit then fetches the given information and stores them in its on-chip channel database.

The first transmit descriptor is fetched from system memory and stored in the chip internal channel database the first time the transmit buffer requests data for a channel. It contains a pointer to the data buffer, the length of the data section as well as a pointer to the next transmit descriptor. After the first descriptor is stored internally a 'Transmit Command Complete' interrupt vector is generated.

Data transfers are requested as long as the number of empty locations is below a programmable refill threshold. The number of empty locations is reported from the transmit buffer to the data management unit. Task of the data management unit is to

Functional Description

calculate the number of bytes that can be loaded from the data section based on the NO field of the transmit descriptor and to compare this with the number of bytes requested by the transmit buffer.

Depending on the bit field NO in the transmit descriptor several read accesses must be performed by the data management unit. It stops serving the request as soon as the requested amount of data was transferred to the transmit buffer, when a Frame End bit (FE) in the processed transmit descriptor is set or when the channel was aborted using a 'Transmit Abort' command. Serving the request can also be suspended, when the programmed transmit burst length (CONF3.TPBL) is reached. All these events may result in open transmit buffer locations, but the data management unit stores this information as open requests in the channel database and processes these requests continuously.

The data management unit alternately serves requests issued by the transmit buffer or open requests stored in its internal channel database. If there are open requests for a channel, data transmission will be initiated. The procedure is the same as described above. It stops, if the requested amount of data is served or when the FE bit field is set.

If a transmit descriptor has its FE bit set and all data of the data section is moved to the transmit buffer, the data management unit serves requests of further channels or looks for open requests in its database. Therefore open requests from other channels are served faster and possible underruns can be avoided. The next transmit descriptor will be retrieved with the next data transfer of the channel.

When the data management unit completed reading a data section associated with a transmit descriptor, it updates the complete (C) bit in the status word of the transmit descriptor if the complete enable (CEN) bit is set. Additionally a 'Transmit Host Initiated' interrupt vector is generated if the THI bit is set in the transmit descriptor. Afterwards the data management unit the MUNICH256FM branches to the next transmit descriptor.

Descriptor hold operation

The data transfer is controlled by the HOLD bit, which is located in the first DWORD of a transmit descriptor. The HOLD bit indicates that the marked descriptor is the last descriptor in a linked list. The data management unit will not branch to the next descriptor until the hold condition is removed or a 'Transmit Abort' command forces the MUNICH256FM to branch to a new linked list.

If the HOLD bit and the frame end bit are set together in a descriptor, the data management unit transfers all data of the belonging data section to the transmit buffer and optionally sets the C-bit in the current transmit descriptor. When a new data transfer is requested (either from the transmit buffer or an open request) the data management unit repolls the descriptor. If the HOLD bit is removed, it will branch to the next transmit descriptor. If the HOLD bit is still set, that channel is suspended for further operation. Following requests from the transmit buffer will not be served, but the number of requested data is stored in the open request registers.

Functional Description

If the HOLD bit is detected in a descriptor and the frame end bit is not set, the data management unit will transfer all data of the belonging data section to the transmit buffer. Afterwards it generates a 'Hold Caused Transmit Abort' interrupt vector in order to inform the host CPU about the erroneous descriptor structure. In PPP and HDLC mode the abort status is propagated to the transmit buffer and the protocol machine, so that a abort sequence is sent on the serial side. In TMA mode the data management unit generates a 'Hold Caused Transmit Abort' interrupt vector every time it recognizes the HOLD bit. Then it reads the transmit descriptor once more. If the HOLD bit is removed it branches to the next transmit descriptor and proceeds with normal operation. Otherwise, when the HOLD bit is still set, the channel is suspended for further operation and an internal poll bit is set. Following requests from the transmit buffer will not be served, but the number of requested data is stored in the open request register.

The host CPU can remove the hold condition, when the next transmit descriptor is available in system memory. Therefore the CPU has to execute a 'Transmit Hold Reset' command, which will reactive the channel. When the transmit buffer requests a new data transfer or when open request are stored in the on-chip database the data management unit repolls the transmit descriptor and checks the HOLD bit again. If the HOLD bit is removed it branches to next transmit descriptor.

If the CPU issues a 'Transmit Hold Reset' command and does not remove the HOLD bit (erroneous programming), no action will take place. Nevertheless, the CPU always has to issue a 'Transmit Hold Reset' command when it removes the HOLD bit in a descriptor, no matter the data management unit has already seen the HOLD bit or not.

4.3.6 Byte Swapping

The MUNICH256FM operates per default as a little endian device. To support integration into big endian environments, the data management unit provides an internal byte swapping mechanism, which can be enabled via bit CONF1.LBE.

The big endian swapping applies only to the data section pointed to by the receive and transmit descriptors in the shared memory.

Note: Byte swapping only effects the organization of packet data in system memory. All internal registers, as well as the descriptors, address pointers or interrupt vectors are handled with little endian byte ordering.

Table 4-3 Example for little/big Endian with BNO = 3

| BNO | Little Endian | | | | Big Endian | | | |
|-----|---------------|--------|--------|--------|------------|--------|--------|---|
| 3 | - | Byte 2 | Byte 1 | Byte 0 | Byte 0 | Byte 1 | Byte 2 | - |

Table 4-4 Example for little big Endian with BNO = 7

| BNO | Little Endian | | | | Big Endian | | | |
|-----|---------------|--------|--------|--------|------------|--------|--------|-------|
| 7 | Byte3 | Byte 2 | Byte 1 | Byte 0 | Byte 0 | Byte 1 | Byte 2 | Byte3 |
| | - | Byte 6 | Byte 5 | Byte 4 | Byte 4 | Byte 5 | Byte 6 | - |

4.3.7 Transmission Bit/Byte Ordering

Data is transmitted beginning with byte zero in increasing order. Vice versa data received is stored starting with byte zero. The position of byte zero depends on the selected endian mode.

Each byte itself consists of eight bits starting with bit zero (LSB) up to bit seven (MSB). Data on the serial line is transmitted starting with the LSB. The first bit received is stored in bit zero.

4.4 Buffer Management

4.4.1 Internal Receive Buffer

The internal receive buffer provides buffering of frame data and status between the protocol handler and the receive data management units. Internal buffers are essential to avoid data loss due to the PCI bus latency, especially in the presence of multiple devices on the same PCI bus, and to enable a minimized bus utilization through burst accesses.

The incoming data from the protocol handler is stored in a receive central buffer shared by all the 256 channels. The buffer is written by the protocol handler every time a complete DWORD is ready or the last byte of a frame has been received. Each channel has an individual programmable threshold code, which determines after how many DWORDs a data transfer into the shared memory is generated. The threshold therefore defines the maximum burst length for a particular channel in receive direction. A data transfer is also requested as soon as a frame end has been reached. Programming the burst length to be greater than 1 DWORD avoids too frequent accesses to the PCI bus, thereby optimizing use of this resource.

For real time channels with lowest possible latency (example: constant bit rate) a value of one DWORD can be selected for the burst length.

The total size of the internal receive buffer is 12 kByte. If all the 256 channels are active, the average burst threshold should be programmed with 8 DWORDs, so that 4 DWORDs are available on the average to compensate for PCI latency and avoid data loss. However if less than 256 channels are active or if only 64 KBit/s channels are used, the burst threshold may be programmed to a higher value. In other words, the sum of all channel thresholds shall not exceed the maximum receive buffer locations.

In order to prevent an overload condition from one particular channel (e.g. receiving only small or invalid frames), the receive buffer provides the capability to delete frames which are smaller or equal than a programmable threshold. All frames that have been dropped will be counted and an interrupt vector will be generated as soon as a programmable threshold has been reached. The actual value of the counter can be read in the small frame dropped counter register.

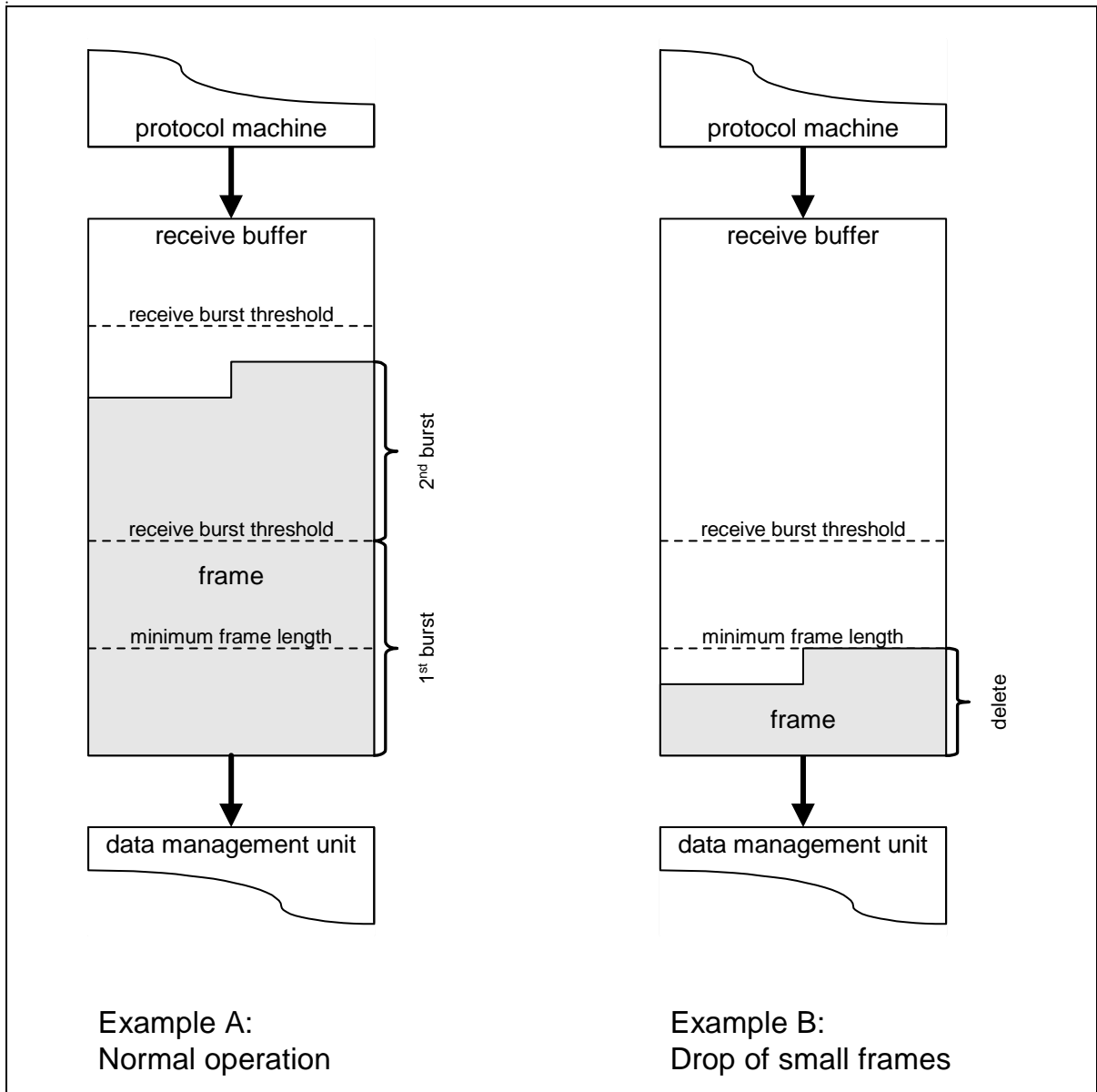


Figure 4-7 Receive Buffer Thresholds

For performance monitoring the receive buffer provides the capability to monitor the receive buffer utilization and to generate interrupts when certain fill thresholds have been reached.

4.4.2 Internal Transmit Buffer

The internal transmit buffer with a total size of 32 kByte stores protocol data before it is processed by the protocol machine. The transmit buffer is essential to ensure that enough data is available during transmission, since PCI latency and usage of multiple

Functional Description

channels limit access to system memory for a particular channel. A programmable transmit buffer size and two programmable threshold are configurable by the host CPU for each channel.

Note: The sum of both thresholds must be smaller than the transmit buffer size of a particular channel.

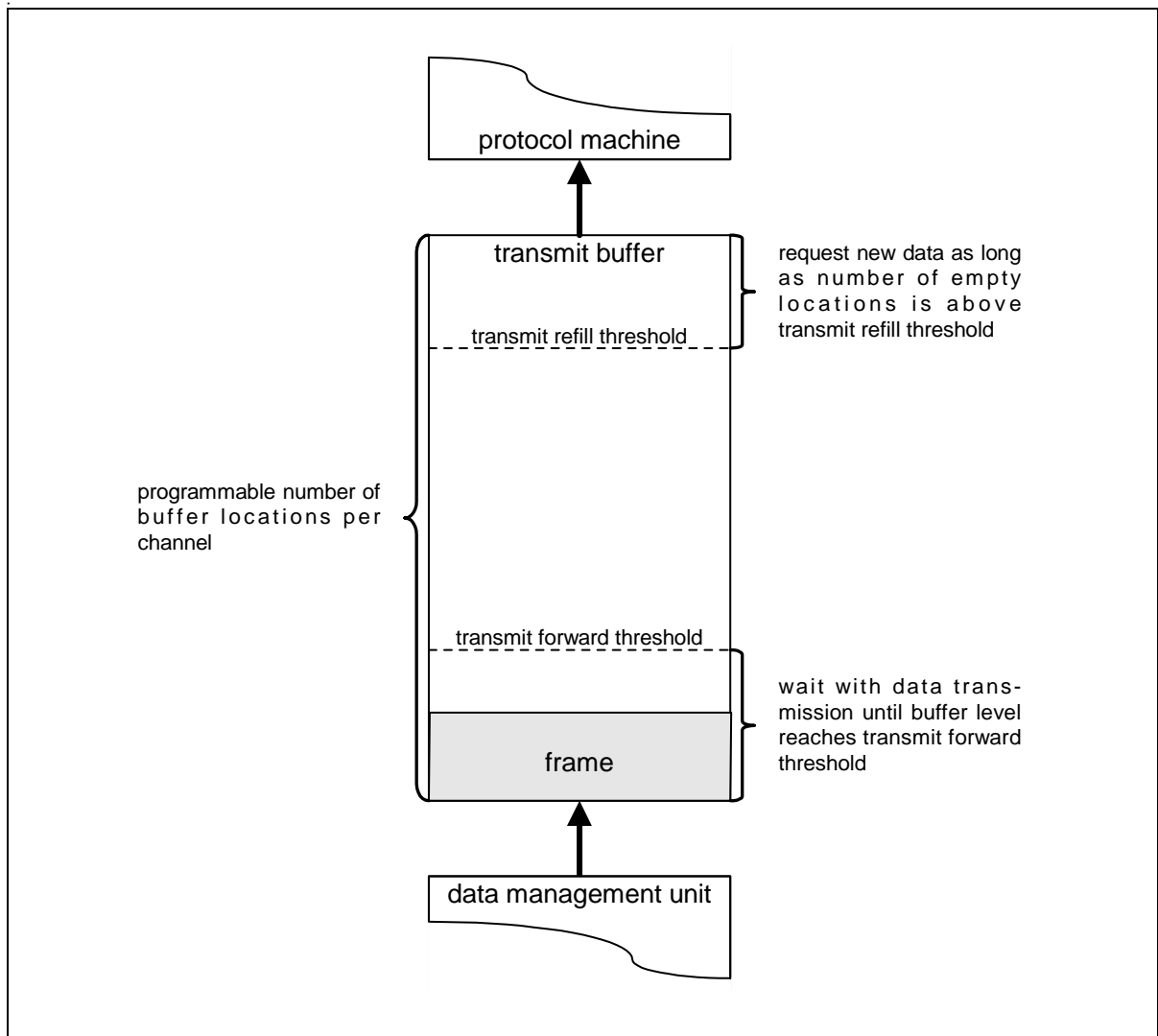


Figure 4-8 Transmit Buffer Thresholds

The threshold values have the following effect:

- Data belonging to one channel stored in the internal transmit buffer will only be transferred to the protocol machine when the transmit forward threshold is reached or if a complete frame is stored inside the transmit buffer. This mechanism avoids data underrun conditions.

Functional Description

- As long as the amount of data stored in the transmit buffer is below the transmit refill threshold the data management unit will keep filling the buffer by initiating PCI burst transfers.

Note: Since there is a delay between the time the transmit buffer requests data from the data management unit and the time the data management unit serves the request, the actual number of empty locations may be higher than the transmit refill threshold. To determine the maximum PCI burst length an additional parameter is available which limits these requests up to a maximum of 64 DWORDS.

4.5 Protocol Description

The protocol machines provide protocol handling for up to 256 channels. The protocol machines implement 4 modes, which can be programmed independently for each channel: HDLC, bit-synchronous PPP, octet-synchronous PPP and transparent mode A.

The configuration of each logical channel is programmed via the PCI bus and will be stored inside the protocol machines. Furthermore the current state for the protocol processing (CRC check, 1 bit count,...) is also stored inside the protocol machines.

Each protocol machine (receive, transmit) handles a maximum of 256 channels and a maximum aggregate bit rate of up to 45 Mbit/s.

4.5.1 HDLC Mode

| | | | | | |
|--------------------------|--------------------------|--------------------------|-------------------------------|--------------------------|--------------------------|
| Flag 0111 1110 | Address 8 bits | Control 8 bits | Information ←0 Bits | CRC 16/32 bits | Flag 0111 1110 |
|--------------------------|--------------------------|--------------------------|-------------------------------|--------------------------|--------------------------|

Figure 4-9 HDLC Frame Format

The frame begin and frame end synchronization is performed with the flag character $7E_H$. Shared opening and closing flag is supported in receive direction and can be programmed in the channel configuration register for transmit direction. Shared '0' bit between two flags is only supported in receive direction. Interframe time-fill can be programmed to either flag $7E_H$ or FF_H indicating idle.

In receive operation, prior to FCS computation, any '0' bit that directly follows five contiguous '1' bits is discarded. When closing flag is recognized, a CRC check, octet boundary check, MFL (maximum frame length) check, a short frame check and an additional small frame check are performed. Short frames have less than 4 octets if CRC16 is used or less than 6 octets if CRC32 is used. An aborted frame is recognized if 7 or more '1's are received.

In transmit operation after the CRC computation a '0' bit is inserted after every sequence of five contiguous '1' bits. When frame end is indicated in the belonging transmit descriptor the calculated CRC is transmitted and a flag is generated. If an underrun occurs in the internal transmit buffer (because of PCI latency e.g.) an abort sequence with 7 '1's is transmitted and an underrun interrupt is generated. The abort sequence is also generated if the host CPU resets or aborts a channel during the transmission of a frame.

An invert option is provided to invert all the data output or data input between serial line and protocol machines or vice versa.

The following CRC modes are supported:

- 16 bit CRC $1+x^5+x^{12}+x^{16}$

Functional Description

- 32 bit CRC $1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}$

Optionally CRC transfer and check can be disabled.

4.5.2 Bit Synchronous PPP with HDLC Framing Structure

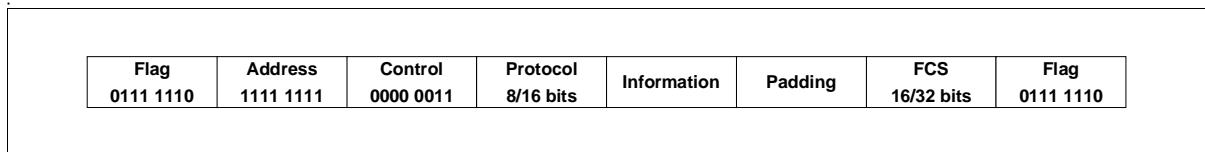


Figure 4-10 Bit Synchronous PPP with HDLC Framing Structure

Same as HDLC. The handling of the abort sequence differs from that in HDLC mode. If $7E_H$ is programmed as interframe time fill character, the abort sequence consists of 7 “1”s. If FF_H is programmed as interframe time fill character, the abort sequence consists of 15 “1”s.

The same programmable parameters as in HDLC mode apply to bit synchronous PPP.

4.5.3 Octet Synchronous PPP

This mode uses a frame structure similar to the bit synchronous PPP mode. The frame begin and end synchronization is performed with the flag character ($7E_H$). Use of a shared opening and closing flag is supported if programmed in the channel configuration register. Use of a shared ‘0’ bit between two flags is not supported. A 16 or 32 bit CRC is computed over all service data read from the transmit buffer and appended to the end of the frame.

The octet synchronous PPP mode uses octet stuffing instead of ‘0’ bit stuffing in order to replace control characters used by intervening hardware equipment. This allows transparent transmission and also recognition and removal of spurious characters inserted by such equipment.

A 32 bit per channel asynchronous control character map (ACCM) specifies characters in the range 00_H-1F_H to be stuffed/destuffed in service data and FCS field. In addition, the DEL control character and any of 4 ACCM extension characters stored in a programmable 32 bit register can be selected for character stuffing/destuffing. When a character specified to be mapped is found in service data or the FCS field, it is replaced by a 2 octet sequence consisting of $7D_H$ (Control Escape) followed by the character EXORed with 20_H (e.g. 13_H is mapped to $7D_H 33_H$). In addition to the per channel specification of characters to be mapped, the control escape sequence $7D_H$ and $7E_H$ in the service data stream are always mapped. Opening and closing flags are not affected.

The abort sequence consists of the control escape character followed by a flag character $7E_H$ (not stuffed).

Between two frames, the interframe time fill character is always $7E_H$.

If in the transmit direction a data underrun occurs during transmission of a frame and the frame has not finished, an abort sequence is automatically sent (escape character followed by a flag) and an underrun interrupt vector will be generated. If the transmit buffer indicates an empty condition for a channel between two frames (idle or interframe fill), the protocol machine will continue to send interframe time fill characters. Also an abort sequence will be generated if a channel is reset or an abort command is issued during transmission of a frame.

The following CRC modes are supported:

- 16 bit CRC $1+x^5+x^{12}+x^{16}$
- 32 bit CRC $1+x+x^2+x^4+x^5+x^7+x^8+x^{10}+x^{11}+x^{12}+x^{16}+x^{22}+x^{23}+x^{26}+x^{32}$

CRC computation/check or removing can be disabled.

4.5.4 Transparent Mode

When programmed in transparent mode, the protocol machine performs fully transparent data transmission/reception without HDLC framing, i.e. without

- Flag insertion/removing
- CRC generation/CRC check
- Bit stuffing/destuffing (0 bit insertion/removal).

An option 'Transparent Mode Pack' is provided to support subchanneling. If subchanneling is used (logical channels of less than 64 kbit/s), masked bits in the protocol data are set high and each bit in shared memory maps directly to enabled (not masked) bits on the serial line. Otherwise they contain protocol data, that is each byte in shared memory maps directly to a time slot.

A programmable transparent flag can be programmed which will be inserted between payload data or is removed during reception of a payload data.

An invert option is provided to invert the outgoing or incoming data stream.

4.6 T1 Framer and FDL Function

The T1 framer includes frame alignment, CRC-6 check/generation, facility data link (FDL) support and bit error rate test. Three modes can be programmed for each T1 link: F4, ESF (F24), SF (F12).

4.6.1 4-Frame Multiframe

The allocation of the FT bits (bit 1 of frames 1 and 3) for frame alignment signal is shown in **Table 4-5**.

The FS bit may be used for signaling.

Remote alarm (yellow alarm) is indicated by setting bit(2) to '0' in each channel.

Table 4-5 4-Frame Multiframe Structure.

| Frame Number | F _T | F _S |
|--------------|----------------|----------------|
| 1 | 1 | |
| 2 | – | Service bit |
| 3 | 0 | |
| 4 | – | Service bit |

Synchronization Procedure

For multiframe synchronization, the terminal framing bits (FT bits) are observed. The synchronous state is reached if at least one terminal framing candidate is definitely found, or the synchronizer is forced to lock onto the next available candidate (RCMDR.FRS).

4.6.2 ESF Mode

The ESF multiframe consists of 24 consecutive frames. The first bit of each frame (F bit) is used as frame alignment, data link channel and CRC-6 channel (see **Table 4-6**).

Table 4-6 ESF Multiframe Structure

| Frame number | F bits | | | |
|--------------|-----------------------|--------------------------------|----------------|---------------------------------|
| | Superframe bit number | Framing Pattern Sequence (FPS) | Data link (DL) | Cyclic redundancy check (CRC-6) |
| 1 | 0 | - | m | - |
| 2 | 193 | - | - | c1 |
| 3 | 386 | - | m | - |
| 4 | 579 | 0 | - | - |
| 5 | 772 | - | m | - |
| 6 | 965 | - | - | c2 |
| 7 | 1158 | - | m | - |
| 8 | 1351 | 0 | - | - |
| 9 | 1544 | - | m | - |
| 10 | 1737 | - | - | c3 |
| 11 | 1930 | - | m | - |
| 12 | 2123 | 1 | - | - |
| 13 | 2316 | - | m | - |
| 14 | 2509 | - | - | c4 |
| 15 | 2702 | - | m | - |
| 16 | 2895 | 0 | - | - |
| 17 | 3088 | - | m | - |
| 18 | 3281 | - | - | c5 |
| 19 | 3474 | - | m | - |
| 20 | 3667 | 1 | - | - |
| 21 | 3860 | - | m | - |
| 22 | 4053 | - | - | c6 |

Functional Description

| | | | | |
|----|------|---|---|---|
| 23 | 4246 | - | m | - |
| 24 | 4439 | 1 | - | - |

Frame 1 is transmitted first. Bit 1 (most significant bit) of each frame is transmitted first.

4.6.2.1 Multiframe Synchronization Procedure of the Receiver

The F-bit of every fourth frame forms the pattern 001011. This multiframe alignment allows to identify where each particular frame is located within the multiframe in order to extract the cyclic redundancy check code (CRC-6) and the data link information.

In the synchronous state two errors within 4 or 5 framing bits, two or more erroneous framing bits within one ESF multiframe or 4 consecutive errored multiframe will lead to the asynchronous state.

There are two multiframe synchronization modes selectable via RFMR.SSP:

- 0 In the synchronous state, the setting of RCMDR.FRS resets the synchronizer and initiates a new frame search. The synchronous state will be reached again, if there is only one definite framing candidate. In the case of repeated apparent simulated candidates, the synchronizer remains in the asynchronous state.
In asynchronous state, setting bit RCMDR.FRS induces the synchronizer to lock onto the next available framing candidate if there is one. At the same time the internal framing pattern memory will be cleared and other possible framing candidates are lost.
- 1 In the synchronous state, the setting of RCMR.FRS resets the synchronizer and initiates a new frame search. Synchronization is achieved if there is only one definite framing candidate AND the CRC-6 checksum is received without an error. If the CRC-6 check failed on the assumed framing pattern the MUNICH256FM will stay in the asynchronous state, searching for an alternate framing pattern.
In case no alternate framing pattern can be found, setting bit RCMDR.FRS starts a totally new multiframe search. At the same time the internal framing pattern memory will be cleared and other possible framing candidates are lost.

4.6.2.2 CRC-6 Generation / Check according to ITU-T G.706

Generation

In calculating the CRC-6 bits, the F-bits are replaced by binary 1s. All information in the other bit positions will be identical to the information in the corresponding multiframe bit positions.

The CRC-6 bit sequence c1, c2, c3, c4, c5, c5 and c6 calculated on multiframe N is transmitted in multiframe N+1. This CRC polynomial is defined as the remainder after

Functional Description

multiplication by x^6 and then division (modulo 2) by the generator polynomial x^6+x+1 of the polynomial corresponding to multiframe N. The first check bit c_1 is the most significant bit of the remainder; the last check bit c_6 is the least significant bit of the remainder.

Check

At the receiver, the received multiframe, with each F-bit having first been replaced by a binary 1, is acted upon by the multiplication/division process described above. The resulting remainder is compared on a bit-by-bit basis, with the CRC-6 check bits contained in the subsequently received multiframe.

In synchronous state a received CRC-6 error may generate an interrupt status and will increment a CRC-6 counter.

4.6.2.3 Remote Alarm (Yellow Alarm) Generation / Detection

Generation

If $TFMR.AXRA=1$, the remote alarm sequence will be automatically sent in the outgoing data stream when the receiver is in asynchronous state (FRS.LFA bit is set). Remote Alarm is also sent unconditionally when $TCMDR.XRA=1$. ESF RA is sent by repeating the pattern '1111 1111 0000 0000' in the Data Link (DL).

Detection

Remote Alarm (yellow alarm) is detected and flagged with bit FRS.RRA when the pattern '1111 1111 0000 0000' is received in the DL bits if $RFMR.SRAF=0$. If $RFMR.SRAF=1$, yellow alarm is detected when every bit2 of each time slot is 0. If $RFMR.RRAM$ is set, Remote Alarm can be detected even in the presence of BER 1/1000. FRS.RRA will be reset automatically when the alarm condition is no longer detected.

4.6.2.4 Facility Data Link

The Facility Data Link (FDL) contains bit oriented messages (priority or command/response) or HDLC-based message oriented signals that are processed by a HDLC machine. Each T1 port has its dedicated FDL controller. In HDLC mode CRC16 is supported. Additionally one or two byte address comparison is supported.

Note: CAS - BR (Channel Associated Signalling - bit robbing) is not supported. The protocol machines support access to 56 kBit/s or 64 kBit/s data channels with their bit masking function. If CCS (Common Channel Signalling) is used, the corresponding channel (usually time slot 24) is handled as a standard data time slot by the HDLC/PPP machine and the data is transferred via the PCI bus.

Functional Description

In transmit and receive direction 64 byte deep FIFOs divided into two pages of 32 bytes are provided for the intermediate storage of data between the HDLC machine and the CPU interface.

Receive Signaling Controller

Each of the signaling controllers may be programmed to operate in various signaling modes. The MUNICH256FM will perform the following signaling and data link methods on the DL-Channel of the ESF format:

- **HDLC/SDLC Access**

In case of common channel signaling the signaling procedure HDLC/SDLC will be supported. The signaling controller of the MUNICH256FM performs the flag detection, CRC checking, address comparison and zero bit-removing. Depending on the selected address mode, the MUNICH256FM may perform a 1 or 2 byte address recognition. If a 2-byte address field is selected, the high address byte is compared with two individually programmable values in register RAH. Buffering of receive data is done in the RFIFO. Refer also to **Chapter 4.8.1**.

- **Transparent Access**

In signaling controller transparent mode, fully transparent data reception without HDLC framing is performed, i.e. without flag recognition, CRC checking or bit-stuffing. This allows the user specific protocol variations.

- **Bit Oriented Messages in ESF-DL Channel**

The MUNICH256FM supports the DL-channel protocol for ESF format according to ANSI T1.403 specification or according to AT&T TR54016. The Bit Oriented Message (BOM) receiver may be switched on/off separately. If the MUNICH256FM is used for HDLC formats only, the BOM receiver has to be switched off. If BOM-receiver has been switched on, an automatic switching between HDLC and BOM mode is enabled. If eight or more consecutive ones are detected, the BOM mode is entered. Upon detection of a flag in the data stream, the MUNICH256FM switches back to HDLC-mode. In BOM-mode, the following byte format is assumed (the left most bit is received first).

11111110xxxxx0

The MUNICH256FM uses the FF_H byte for synchronization, the next byte is stored in RFIFO (first bit received: LSB) if it starts and ends with a '0'. Bytes starting or ending with a '1' are not stored. If there are no 8 consecutive one's detected within 32 bits and the MUNICH256FM is currently in the BOM mode, an interrupt is generated. However, byte sampling is not stopped.

Transmit Signaling Controller

Similar to the receive signaling controller the same signaling method is provided. The MUNICH256FM will perform the following signaling and data link methods on the DL-channel of the ESF format:

Functional Description

- **HDLC access**

The transmit signaling controller of the MUNICH256FM performs the FLAG generation, CRC generation, zero bit-stuffing and programmable IDLE code generation. Buffering of transmit data is done in the 2x32 byte deep transmit FIFO. The signaling information will be internally multiplexed with the data applied to the outgoing ports.

- **Transparent/BOM mode**

In signaling controller transparent mode, fully transparent data transmission without HDLC framing is performed. Optionally the MUNICH256FM supports the continuous transmission of the XFF.XFIFO contents with a maximum of 32 bytes.

Operating in HDLC or BOM mode “flags” or “idle” may be transmitted as interframe time-fill.

4.6.3 SF Mode

The SF multiframe consists of 12 consecutive frames. The first bit of each frame (F-bit) the MUNICH256FM is used as frame alignment (see following table).

Table 4-7 SF Multiframe Structure

| Frame number | F-bits | | |
|--------------|-----------------------|-----------------------|------------------------|
| | Superframe bit number | Terminal Framing (Ft) | Signaling Framing (Fs) |
| 1 | 0 | 1 | - |
| 2 | 193 | - | 0 |
| 3 | 386 | 0 | - |
| 4 | 579 | - | 0 |
| 5 | 772 | 1 | - |
| 6 | 965 | - | 1 |
| 7 | 1158 | 0 | - |
| 8 | 1351 | - | 1 |
| 9 | 1544 | 1 | - |
| 10 | 1737 | - | 1 |
| 11 | 1930 | 0 | - |
| 12 | 2123 | - | 0 |

The Fs-bits are used to get a higher synchronization probability but no CAS - BR (Channel Associated Signalling - bit robbing) is supported. Only frame alignment is provided in this mode.

4.6.3.1 Synchronization Procedure of the Receiver

In the synchronous state terminal framing (Ft-bits) and multiframing (Fs-bits) are observed, independently. Further reaction on framing errors depends on the selected synchronization/resynchronization procedure (via bit RFMR0.SSP):

- 0 Terminal frame and multiframe synchronization are combined. Two errors within 4/5/6 Ft-bits or two errors within 4/5/6 in Fs-bits (via bits RFMR.SSC) will lead to the asynchronous state for terminal framing and multiframing. Additionally to the bit FRS.LFA, loss of multiframe alignment is reported via bit FRS.LMFA. The resynchronization procedure starts with synchronizing upon the terminal framing. If the pulseframing has been regained, the search for

Functional Description

multiframe alignment is initiated. Multiframe synchronization has been regained after two consecutive correct multiframe patterns have been received.

- 1 Terminal frame and multiframe synchronization are separated. Two errors within 4/5/6 terminal framing bits will lead to the same reaction as described above for the 'combined' mode. Two errors within 4/5/6 multiframe bits will lead to the asynchronous state only for the multiframe. Loss of multiframe alignment is reported via bit FRS.LMFA. The state of terminal framing is not influenced. Now, the resynchronization procedure includes only the search for multiframe alignment. Multiframe synchronization has been regained after two consecutive correct multiframe patterns have been received.

4.6.3.2 Remote Alarm (Yellow Alarm) Generation / Detection

There are two possibilities of remote alarm (yellow alarm) indication:

- Bit 2 = '0' in each time slot of the frame, selected with bit R/TFMR.SRAF = 0
- The last bit of the multiframe alignment signal (bit 1 of frame 12) changes from '0' to '1', selected with bit R/TFMR.SRAF = 1.

Generation

If TFMR.AXRA=1, the remote alarm sequence will be automatically sent in the outgoing data stream when the receiver is in asynchronous state (FRS.LFA bit is set). Remote Alarm is also sent unconditionally when TCMDR.XRA = 1.

Detection

Remote alarm (yellow alarm) is detected and flagged with bit FRS.RRA which will be reset automatically when the alarm condition is no longer detected.

4.6.4 Common Features for SF and ESF

4.6.4.1 AIS (Blue Alarm) Generation/Detection

Generation

The alarm indication signal is an all one unframed signal and will be transmitted if enabled via bit TCMDR.XAIS.

Detection

The detection of AIS is done, if 2 or less '0's are detected in a multiframe. This condition is flagged by bit FRS.AIS. AIS detection can also only be enabled in asynchronous state by bit RFMR0.AIS3. In this case AIS is indicated if three or less zeros within a time interval of 12 frames (in SF mode), or if five or less zeros within a time interval of 24 frames (ESF mode) are detected in the received bit stream.

4.6.4.2 Loss of Signal (Red Alarm) Detection

The MUNICH256FM can be programmed to satisfy the different definitions for detecting Loss of Signal (LOS) alarms in ITU-T G.775 and AT&T TR54016. Loss of signal is indicated by a flag in the receive framer's status register (FRS.LOS). In addition, a 'Loss of Signal Status' interrupt vector is generated, if not masked.

LOS detection and recovery conditions are set by a flag RFMR.LOSR and the two parameters PCD and PCR.

Detection

'Loss of Signal' alarm will be generated, if the incoming data stream has no pulses (no '1') for a certain number N of consecutive bits. 'No pulse' in the receive interface means a logical zero octet on receive data inputs. The number N can be set via register PCD and is calculated as $8 \cdot (PCD + 1)$.

Recovery

The recovery procedure starts after detecting a logical '1' in the received bit stream. The value via register PCR defines the number of pulses, which must occur during the time interval $8 \cdot (PCD + 1)$, to clear the LOS alarm.

Additionally, if selected via RFMR.LOSR, any pulse density violation resets the measurement interval. I.e. in addition to the basic pulse density required for recovery, a density of at least N '1's in every N+1 octets ($0 < N < 24$) is required during $8 \cdot (PCD + 1)$ bit intervals.

4.6.4.3 In-Band Loop Generation and Detection

The MUNICH256FM generates and detects a framed or unframed in-band loop up/actuate (00001) and down/deactuate (001) pattern according to ANSI T1.403 even in the presence of bit error rates as high as 1/100. Replacing the transmit data with the in-band loop codes is done by TCMDR.XLD / XLU for actuate or deactuate loop code.

The CPU must reset this bit to 0 for normal operation (no loop-back code). The MUNICH256FM also offers the ability to generate and detect a flexible in-band loop up/actuate and down/deactuate pattern. The loop up and down pattern is individual programmable in the Loop Code Register from 5 to 8 bits in length.

Status and interrupt-status bits will inform the user whether Loop Actuate- or Deactuate code was detected, but the CPU must activate the loop-back.

4.6.4.4 Pulse Density Detection

The framer examines the receive data stream of each port on the pulse density requirement defined by ANSI T1. 403. More than 15 consecutive zeros or less than N ones in each and every time window of $8(N+1)$ data bits, where $N=23$ will be detected. Violations of these rules are indicated by setting the status bit FRS.PDEN. Moreover the PDEN bit in the interrupt vector will be set.

4.6.4.5 Error Performance Monitoring

The MUNICH256FM supports the error performance monitoring by detecting following alarms in the received data.

- Framing errors
- CRC errors
- Loss of frame alignment
- Loss of signal
- Alarm indication signal

Loss of frame alignment, Loss of signal and AIS are indicated with interrupt status bits. With a programmable interrupt mask (register IMR) all these error events could generate an Errored Second interrupt (ES) if enabled. Additionally a one Second interrupt could be generated to indicate that the ES interrupt has to be read. If the ES interrupt is set the enabled alarm status bits or the error counters have to be examined.

The following counters are implemented in the T1 framer:

- Framing Error Counter: This counter will be incremented when incorrect FT and FS bits in SF mode or incorrect FPS bits in ESF format are received. Framing errors will not be counted during asynchronous state.
- CRC Error Counter (Only ESF mode): The counter will be incremented when a multiframe has been received with a CRC error. CRC errors will not be counted during asynchronous state.

Functional Description

- Errored block counter: This counter will be incremented, if a multiframe has been received with framing errors or CRC errors (ESF only).

Clearing and updating of the counters is done according to bit RFMR1.ECM. If this bit is reset, the error counter is permanently updated. Reading of actual error counter status is always possible. The error counters are reset by reading the corresponding status register. If RFMR1.ECM is set, every second the error counter will be latched and then automatically reset. The latched error counter state should be read within the next second.

4.6.4.6 Pseudo-random Bit Sequence Generator and Monitor

A Pseudo-random bit sequence (PRBS) generator and monitor according to ITU O.151 can be activated for one particular logical channel. The PRBS pattern type can be selected as $2^{15}-1$ or $2^{20}-1$ via R/TPRBSC.PRP. Moreover, the number of the time slots which should be used for PRBS can be defined in R/TPTSL register.

Additionally a fixed pattern can be programmed via registers R/TFPR0 and R/TFPR1 with length up to 32 bit to be defined in R/TPRBSC.FPL.

The PRBS monitor searches synchronization on the inverted and non-inverted PRBS pattern. The current synchronization status is reported in status and interrupt status registers. Each PRBS bit error will increment an error counter. An additional counter will accumulate the total number of received bits. Synchronization will be reached within 400 ms with a probability of 99.9% and a BER of 1/10.

4.7 E1 Framing and Signaling

The operating mode of the MUNICH256FM is selected by programming the carrier data rate and characteristics, multiframe structure, and signaling scheme.

The MUNICH256FM implements the standard framing structures for E1 or PCM 30 (CEPT, 2048 Kbit/s) carriers. The internal HDLC controller supports signaling procedures like signaling frame synchronization/synthesis and signaling alarm detection in all framing formats.

Summary of E1- Framing Modes:

- Doubleframe format according to ITU-T G. 704.
- Multiframe format according to ITU-T G. 704
CRC-4 processing according to ITU-T G. 706.
- Multiframe format with CRC-4 to non CRC-4 interworking according to ITU-T G. 706.

After reset, the MUNICH256FM is switched into doubleframe format automatically. Switching between the framing formats is done via bit T/RFMR.FM

4.7.1 Doubleframe Format

The framing structure is defined by the contents of time-slot 0 (refer to **Table 4-8**).

Table 4-8 Allocation of Bits 1 to 8 of Time slot 0

| Alternate Frames | Bit Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---------------|------------------------|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|---|
| Frame Containing the Frame Alignment Signal | Si | 0 | 0 | 1 | 1 | 0 | 1 | 1 | |
| | ¹⁾ | Frame Alignment Signal | | | | | | | |
| Frame not Containing the Frame Alignment Signal | Si | 1 | A | S _{a4} | S _{a5} | S _{a6} | S _{a7} | S _{a8} | |
| | ¹⁾ | ²⁾ | ³⁾ | ⁴⁾ | | | | | |

- 1) Si-bits:
Reserved for international use.-They are fixed to '1'.
- 2) Fixed to '1'. Used for synchronization.
- 3) Remote alarm indication:
In undisturbed operation '0'; in alarm condition '1'.
- 4) S_a-bits:
Reserved for national use. If not used, they should be fixed at '1'. Access to received information via registers RSAW1-3. Transmission via registers XSAW1-XSAW3. HDLC signalling in bits S_{a4} - S_{a8} is selectable.

4.7.1.1 Synchronization Procedure of the Receiver

Synchronization status is reported via bit FRS.LFA. Framing errors are counted by the Framing Error Counter (FEC). Asynchronous state is reached after detecting 3 or 4 consecutive incorrect FAS words or 3 or 4 consecutive incorrect service words (bit 2 = 0 in time-slot 0 of every other frame not containing the frame alignment word), the selection is done via bit RFMR.SSC. Additionally, the service word condition can be disabled. When the framer lost its synchronization an status bit FRS.LFA is generated. In asynchronous state, counting of framing errors will be stopped.

The resynchronization procedure starts automatically after reaching the asynchronous state. Additionally, it may be invoked user controlled via bit RCMDR.FRS (Force Resynchronization: the FAS word detection is interrupted. In connection with the above conditions this will lead to asynchronous state. After that, resynchronization starts automatically).

Synchronous state is established after detecting:

- a correct FAS word in frame n ,
- the presence of the correct service word (bit 2 = 1) in frame $n + 1$,
- a correct FAS word in frame $n + 2$.

If the service word in frame $n + 1$ or the FAS word in frame $n + 2$ or both are not found searching for the next FAS word will be start in frame $n + 2$ just after the previous frame alignment signal.

Reaching the asynchronous state causes the removal of FSR.LFA and additionally an interrupt vector with LFA bit reset (if not masked). Undisturbed operation starts with the beginning of the next doubleframe.

4.7.1.2 A-bit Access

If the MUNICH256FM detects a remote alarm indication in the received data stream the interrupt status bit FRS.RRA will be set.

By setting TFMR.AXRA the MUNICH256FM automatically transmits the remote alarm bit = 1 in the outgoing data stream if the receiver detects a loss of frame alignment FRS.LFA = 1. If the receiver is in synchronous state FRS.LFA = 0 the remote alarm bit will be reset.

4.7.1.3 S_a -bit Access

The MUNICH256FM allows access to the S_a -bits via registers RSAW1-3 and XSAW1-3.

4.7.2 CRC-4 Multiframe

The multiframe structure shown in **Table 4-9** is enabled by setting TFMR.FM for the transmitter and RFMR.FM for the receiver.

Multiframe : 2 submultiframes = 2 × 8 frames

Frame alignment: refer to **Chapter 4.7.1** Doubleframe Format

Multiframe alignment: bit 1 of frames 1, 3, 5, 7, 9, 11 with the pattern '001011'

CRC bits : bit 1 of frames 0, 2, 4, 6, 8, 10, 12, 14

CRC block size: 2048 bit (length of a submultiframe)

CRC procedure: CRC-4, according to ITU-T G.704, G.706

Table 4-9 CRC-4 Multiframe Structure

| | Sub-Multiframe | Frame Number | Bits 1 to 8 of the Frame | | | | | | | |
|------------|----------------|--------------|--------------------------|---|-----------------|-----------------|------------------|------------------|-----------------|-----------------|
| | | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Multiframe | I | 0 | C ₁ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 1 | 0 | 1 | A | S _{a4} | S _{a5} | S _{a61} | S _{a7} | S _{a8} |
| | | 2 | C ₂ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 3 | 0 | 1 | A | S _{a4} | S _{a5} | S _{a62} | S _{a7} | S _{a8} |
| | | 4 | C ₃ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 5 | 1 | 1 | A | S _{a4} | S _{a5} | S _{a63} | S _{a7} | S _{a8} |
| | | 6 | C ₄ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | 7 | 0 | 1 | A | S _{a4} | S _{a5} | S _{a64} | S _{a7} | S _{a8} | |
| | II | 8 | C ₁ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 9 | 1 | 1 | A | S _{a4} | S _{a5} | S _{a61} | S _{a7} | S _{a8} |
| | | 10 | C ₂ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 11 | 1 | 1 | A | S _{a4} | S _{a5} | S _{a62} | S _{a7} | S _{a8} |
| | | 12 | C ₃ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| | | 13 | E | 1 | A | S _{a4} | S _{a5} | S _{a63} | S _{a7} | S _{a8} |
| | | 14 | C ₄ | 0 | 0 | 1 | 1 | 0 | 1 | 1 |
| 15 | | E | 1 | A | S _{a4} | S _{a5} | S _{a64} | S _{a7} | S _{a8} | |

E Spare bits for international use. E bits are replaced by XSP.XS13 and XSP.XS15 or automatic transmission for submultiframe error indication.

S_a Spare bits for national use. S_a-bit access via registers RSAW1-3 and XSAW1-3 is provided. HDLC-signaling in bits S_{a4} - S_{a8} is selectable.

C₁ ... C₄ Cyclic redundancy check bits.

A Remote alarm indication. Automatic transmission of the A-bit is selectable.

The CRC procedure is automatically invoked when the multiframe structure is enabled. CRC errors in the received data stream are counted by the 16 bit CRC Error Counter CEC (one error per submultiframe, maximum).

Additionally a CRC error interrupt vector with CRC set can be generated if enabled.

4.7.2.1 Synchronization Procedure of the Receiver

Multiframe alignment is assumed to have been lost if doubleframe alignment has been lost (flagged at status bits FRS.LFA and FRS.LMFA). Either edge of these bits will cause an LFA interrupt.

The multiframe resynchronization procedure starts when Doubleframe alignment has been regained which is indicated by a FAS interrupt vector. For Doubleframe synchronization refer to **Chapter 4.7.1**. It may also be invoked by the user by setting bit RFMR.FRS for complete doubleframe **and** multiframe resynchronization.

The CRC checking mechanism will be enabled after the first correct multiframe pattern has been found. However, CRC errors will not be counted in asynchronous state.

The multiframe synchronous state is established after detecting two correct multiframe alignment signals at an interval of $n \times 2$ ms ($n = 1, 2, 3 \dots$). The loss of multiframe alignment flag FRS.LMFA will be reset. Additionally a multiframe alignment status interrupt MFAS is generated on the falling edge of bit FRS.LMFA.

Automatic Force Resynchronization

In addition, a search for Doubleframe alignment is automatically initiated if two multiframe pattern with a distance of $n \times 2$ ms have not been found within a time interval of 8 ms after doubleframe alignment has been regained. The new search for frame alignment will be started just after the previous frame alignment signal.

CRC-4 Interworking Mode

CRC-4 interworking is implemented according to ITU-T G.706 Appendix B. For operational description refer to **Figure 4-11**.

4.7.2.2 CRC-4 Performance Monitoring

In the synchronous state checking of multiframe pattern is disabled. However, with bit RFMR.ALMF an automatic multiframe resynchronization mode can be activated. If 915 out of 1000 errored CRC submultiframes are found then a false frame alignment will be assumed and a search for double- and multiframe pattern is initiated. The new search for frame alignment will be started just after the previous basic frame alignment signal. The internal CRC-4 resynchronization counter will be reset when the multiframe synchronization has been regained.

4.7.2.3 A-Bit Access

If the MUNICH256FM detects a remote alarm indication (bit 2 in TS0 not containing the FAS word) in the received data stream a RAS interrupt will be generated. With the deactivation of the remote alarm the remote alarm status interrupt with RAS='0' is generated.

By setting TFMR.AXRA the MUNICH256FM automatically transmits the remote alarm bit = '1' in the outgoing data stream if the receiver detects a loss of frame alignment (FRS.LFA = '1'). If the receiver is in synchronous state (FRS.LFA = '0') the remote alarm bit will be reset in the outgoing data stream.

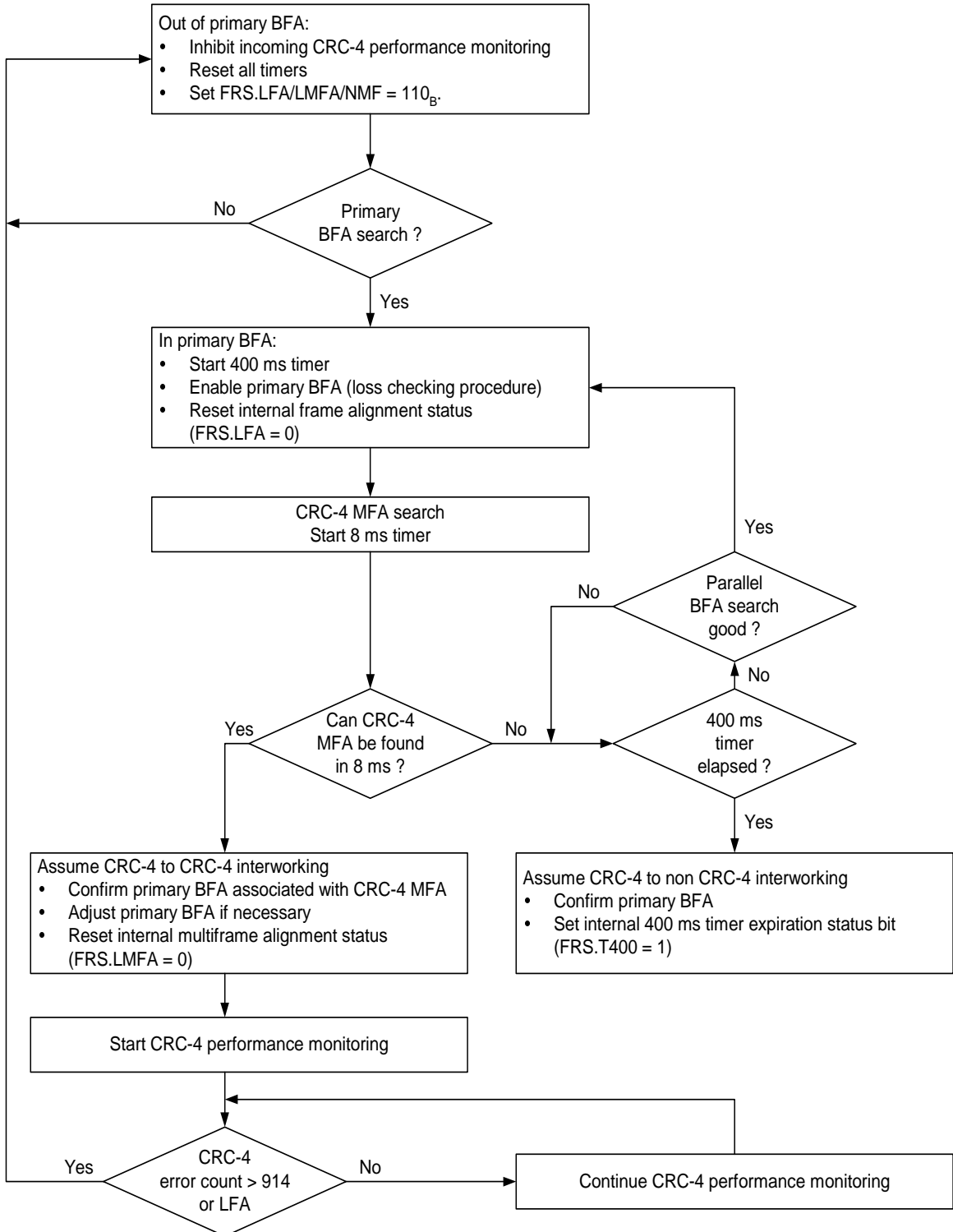


Figure 4-11 CRC-4 Multiframe Alignment Recovery Algorithms

4.7.2.4 S_a -bit Access

Due to signaling procedures using the five S_a -bits ($S_{a4} \dots S_{a8}$) of every other frame of the CRC-4 multiframe structure, two possibilities of access via the microprocessor are implemented.

- The standard procedure, allows reading/writing the S_a -bit registers RSAW1 to RSAW3 and XSAW1 through XSAW3.
Registers RSAW1-3 contains the service word information of the previously received CRC-4 multiframe or 8 doubleframes (bit slots 4-8 of every service word). These registers will be updated on every multiframe. Optionally MUNICH256FM provides the possibility to check the received S_a -data with the S_a -data received earlier. An interrupt vector is generated on S_a -data change in order to reduce microprocessor bus load. With the transmit multiframe begin the contents of this registers XSAW1-3 will be copied into shadow registers. The contents will subsequently sent out in the service words of the next outgoing CRC-4 multiframe (or doubleframes). The TXSA interrupt request that these registers should be serviced. If requests for new information will be ignored, current contents will be repeated.
- The extended access via the receive and transmit FIFOs of the signaling controller. In this mode it is possible to transmit / receive a HDLC frame or a transparent bit stream in any combination of the S_a -bits.

S_a -bit Detection according to ETS 300233

Four consecutive received S_a -bits are checked on the by ETS 300233 defined S_a -bit combinations. The MUNICH256FM can be programmed to detect any bit combination on one S_a -bit out of S_{a4} through S_{a8} . Enabling of specific bit combination can be done via register RCR2.SASSM. A valid S_a -bit combination must occur three times in a row. The corresponding status in register RSAW4 will be set. Register RSAW4 is from type "Clear on Read". With any change of state of the selected S_a -bit combinations a 'SSM Data Valid' interrupt vector will be generated.

During the basic frame asynchronous state updating of register RSAW4 and interrupt vector generation is disabled. In CRC-4 multiframe format the detection of the S_a -bit combinations can be done either synchronous or asynchronous to the submultiframe. In synchronous detection mode updating of register RSAW4 is done in the multiframe synch. state. In asynchronous detection mode updating is independent to the multiframe synchronous state.

S_a -bit Error Indication Counters

The S_a -bit error indication counter CRC1 (16 bits) counts either the received bit sequence 0001_B and 0011_B or two user programmable values defined in register VCRC in every submultiframe on a selectable S_a -bit. In the primary rate access digital section CRC errors are reported from the TE via S_{a6} . Incrementing is only possible in the multiframe synchronous state.

Functional Description

The S_a -bit error indication counter CRC2 (16 bits) counts either the received bit sequence 0010_B and 0011_B or two user programmable values defined in register VCRC in every submultiframe on a selectable S_a -bit. In the primary rate access digital section CRC errors detected at T-reference points are reported via S_{a6} . Incrementing is only possible in the multiframe synchronous state.

4.7.2.5 E-Bit Access

Due to signalling procedures, the E-bits of frame 13 and frame 15 of the CRC-4 multiframe can be used to indicate received errored submultiframes:

no CRC error : E = '1'

CRC error : E = '0'

Standard Procedure

E-bits of the service word are replaced by values of bit XSP.XS13 and XSP.XS15.

Automatic Procedure

Values programmed in register Status information of received submultiframes is automatically inserted in E-bit position of the outgoing CRC-4 Multiframe without any further interventions of the microprocessor.

In the double- and multiframe asynchronous state the E-bits are set to zero. In the multiframe synchronous state the E-bits are processed according to ITU-T G.704.

Submultiframe Error Indication Counter

The Error Bit Counter counts zeros in E-bit position of frame 13 and 15 of every received CRC-4 multiframe. This counter option gives information about the outgoing transmit line if the E-bits are used by the remote end for submultiframe error indication. Incrementing is only possible in the multiframe synchronous state.

4.7.3 Common Features for E1 Doubleframe and CRC-4 Multiframe

4.7.3.1 Error Performance Monitoring and Alarm Handling

Alarm detection and generation

Alarm Indication Signal:

Detection and recovery is flagged by bit FRS.AIS and the 'Alarm Indication Signal Status' interrupt vector. Transmission is enabled via bit TFMR.XAIS.

Loss of Signal:

Detection and recovery is flagged via bit FRS.LOS and a 'Loss of Signal Status' interrupt vector.

Remote Alarm Indication:

Detection and release is flagged by bit FRS.RRA and a 'Remote Alarm Status' interrupt vector. Transmission is enabled via bit TCMDR.XRA.

Table 4-10 Summary of Alarm Detection and Alarm Release

| Alarm | Detection Condition | Clear Condition |
|-------------------------------|--|--|
| Loss of Signal (LOS) | PCD Register No transitions (log. zero octets) in a programmable time interval of 16 - 512 consecutive pulse periods. | PCR Register Programmable amount of ones (1-63) in a progr. time interval of 16 - 512 consecutive pulse periods. The pulse density is fulfilled and no more than 15 or 99 contiguous zeros during the recovery interval are detected. |
| Alarm Indication Signal (AIS) | FMR0.ALM = 0: less than 3 zeros in 250 μ s and loss of frame alignment declared FMR0.ALM = 1: less than 3 zeros in each of two consecutive double frame periods | FMR0.ALM = 0: more than 2 zeros in 250 μ s and frame alignment found FMR0.ALM = 1: more than 2 zeros in each of two consecutive double frame periods |
| Remote Alarm (RRA) | bit 3 = 1 in time-slot 0 not containing the FAS word | set conditions no longer detected. |

Automatic remote alarm access

If the receiver has lost its synchronization a remote alarm could be sent if enabled via TFMR.AXRA to the distant end. The remote alarm bit will be automatically set in the outgoing data stream if the receiver is in asynchronous state (FRS.LFA bit is set). In synchronous state the remote alarm bit will be removed.

Error Counter

The MUNICH256FM framer offers four error counters, each of them has a length of 16 bit. They record framing bit errors, CRC-4 bit errors. Updating the buffer is done in two modes:

- one second boundary
- clear on read

In the one second mode an internal one second timer will update these buffers and reset the counter to accumulating the error events. The error counter can not overflow. Error events occurring during reset will not be lost.

Status: Errored Second

MUNICH256FM supports the error performance monitoring by detecting alarms or error events in the received data.

Loss of frame alignment, including alarm indication signal and loss of signal, as well as CRC errors could generate an Errored Second interrupt if enabled.

Second Timer

An one-second timer interrupt could be internally generated to indicate that the enabled alarm status bits or the error counters have to be checked.

4.7.3.2 Loss of Signal Detection

The MUNICH256FM can be programmed to satisfy the different definitions for detecting Loss of Signal (LOS) alarms in ITU-T G.775 and ETS 300233. Loss of signal is indicated by a flag in the receive framer's status register (FRS.LOS). In addition, a 'Loss of Signal Status' interrupt vector is generated, if not masked.

Detection

'Loss of Signal' alarm will be generated, if the incoming data stream has no pulses (no '1') for a certain number N of consecutive pulse periods. 'No pulse' in the receive interface means a logical zero on receive data inputs. The number N can be set via register PCD and is calculated as $8 \cdot (PCD + 1)$.

Recovery

The recovery procedure starts after detecting a logical '1' in the received bit stream. The value via register PCR defines the number of pulses, which must occur during the time interval $8 \cdot (\text{PCD} + 1)$, to clear the LOS alarm.

4.7.3.3 In-Band Loop Generation and Detection

The MUNICH256FM generates and detects a framed or unframed in-band loop up/actuate (00001) and down/deactuate (001) pattern according to ANSI T1.403 with bit error rates as high as 1/100. Replacing the transmit data with the in-band loop codes is done by TCMDR.XLD / XLU for actuate or deactuate loop code.

The CPU must reset this bit to 0 for normal operation (no loop-back code). The MUNICH256FM also offers the ability to generate and detect a flexible in-band loop up/actuate and down/deactuate pattern. The loop up and down pattern is individual programmable in the Loop Code Register from 5 to 8 bits in length.

Status and interrupt-status bits will inform the user whether Loop Up - or Loop Down code was detected, but the CPU must activate the loop-back.

4.7.3.4 Pseudo-random Bit Sequence Generator and Monitor

A Pseudo-random bit sequence (PRBS) generator and monitor according to ITU O.151 can be activated for one particular logical channel. The PRBS pattern type can be selected as $2^{15}-1$ or $2^{20}-1$ via R/TPRBSC.PRP. Moreover, the number of the time slots which should be used for PRBS can be defined in R/TPTSL register.

Additionally a fixed pattern can be programmed via registers R/TFPR0 and R/TFPR1 with length up to 32 bit to be defined in R/TPRBSC.FPL.

The PRBS monitor searches synchronization on the inverted and non-inverted PRBS pattern. The current synchronization status is reported in status and interrupt status registers. Each PRBS bit error will increment an error counter. An additional counter will accumulate the total number of received bits. Synchronization will be reached within 400 ms with a probability of 99.9% and a BER of 1/10.

Alarm Simulation

Alarm simulation does not affect the normal operation of the device, i.e. all channels remain available for transmission. However, possible 'real' alarm conditions are not reported to the processor or to the remote end when the device is in the alarm simulation mode.

The alarm simulation is initiated by setting different code words in bit field FMR0.SIM. The following alarms are simulated:

- Loss of Signal
- Alarm Indication Signal (AIS)
- Auxiliary pattern
- Loss of pulse frame
- Remote alarm indication
- Framing error counter
- CRC-4 error counter
- E-Bit error counter

Some of the above indications are only simulated if the MUNICH256FM is configured in a mode where the alarm is applicable (e.g. no CRC-4 error simulation when doubleframe format is enabled).

Setting a code word in bit field FMR0.SIM initiates alarm simulation. Error counting and indication will occur while this bit is set. After it is reset all simulated error conditions disappear.

4.8 Signaling Controller Protocol Modes

The signalling controller provides access to the data link and S_a bits of the T1/E1 signal and it provides access to the far end alarm and control channel (FEAC) and the C-bit parity path maintenance data link channel. It operates in HDLC, BOM or automatic modes.

4.8.1 HDLC Mode

In HDLC mode the transmit signaling controller of the MUNICH256FM performs the FLAG generation, CRC generation, zero bit-stuffing and programmable IDLE code generation. Buffering of transmit data is done in the 2x32 byte deep transmit FIFO. The signaling information will be internally multiplexed with the data applied to the outgoing ports and is inserted in or extracted from the DL-Bits in T1 ESF mode or the S_a -bits in E1 modes. Any sequence of S_a -bits can be specified for protocol insertion.

Shared Flags

The closing flag of a previously transmitted frame simultaneously becomes the opening flag of the following frame if there is one to be transmitted. The Shared Flag feature is enabled by setting XCR1.SF.

CRC check

As an option in HDLC mode the internal handling of received and transmitted CRC checksum can be influenced via control bits RCR1.XCRC and XCR1.DISCRC.

- Receive Direction
The received CRC checksum is always assumed to be in the last two bytes of a frame, immediately preceding a closing flag. If RCR1.XCRC is set, the received CRC checksum will be written to RFIFO where it precedes the frame status byte. The received CRC checksum is additionally checked for correctness.
- Transmit Direction
If XCR1.DISCRC is set, the CRC checksum is not generated internally. The checksum has to be provided via the transmit FIFO (XFF.XFIFO) as the last two bytes. The transmitted frame will only be closed automatically with a (closing) flag.

The MUNICH256FM does not check whether the length of the frame, i.e. the number of bytes to be transmitted makes sense or not.

Address comparison

An optional address comparison feature forwards all frames which match a programmable address to the receive FIFO. Frames not matching the address are discarded.

Functional Description

If a 2-byte address field is selected, the high address byte is compared with two individually programmable values defined in register RAH. Similarly, two values can be programmed in register RAL for the low address byte. A valid address is recognized when the high byte and the low byte of the address field correspond to one of the compare values. Thus, the MUNICH256FM can be called (addressed) with 4 different address combinations.

In case of a 1-byte address, RAL will be used as compare registers. The HDLC control field, data in the I-field and an additional status byte are temporarily stored in the receive FIFO.

Preamble Transmission

If enabled, a programmable 8-bit pattern XCR1.PBYTE is transmitted with a selectable number of repetitions after interframe time-fill transmission is stopped and a new frame is ready to be sent out.

Zero Bit Insertion is disabled during preamble transmission. To guarantee correct function the programmed preamble value should be different from Receive Address Byte values.

4.8.2 Transparent Mode

In transparent mode, fully transparent data transmission/reception without HDLC framing is performed, i.e. without FLAG generation/recognition, CRC generation/check, or bit-stuffing. This feature can be profitably used e.g for:

- Specific protocol variations
- Test purposes

Data transmission is always performed out of the transmit FIFO (XFF.XFIFO). In transparent mode receive data is shifted into the receive FIFO without protocol processing.

If the transparent mode is selected, the MUNICH256FM supports the continuous transmission of the contents of the transmit FIFO.

After having written 1 to 32 bytes to transmit FIFO, the command HND via the CMDR register forces the MUNICH256FM to repeatedly transmit the data stored in transmit FIFO to the remote end.

The cyclic transmission continues until a reset command (HND. SRES) is issued or with resetting CMDR.XREP, after which continuous '1'-s are transmitted.

4.8.3 BOM Mode

The signalling controller supports the DL channel protocol for ESF format according to ANSI T1.403 or according to AT&T TR54016. The Bit Oriented Message (BOM) receiver can be switched on or off separately. If the signalling controller is used for HDLC formats

Functional Description

only, the BOM receiver has to be switched off (RCR1.BRAC = '0'). If HDLC and BOM receiver are switched on, an automatic switching between HDLC and BOM mode is done, which depends on the received bit sequence (01111110_B or 11111111_B). If eight or more consecutive ones are detected, the BOM mode is entered automatically. Upon detection of a flag in the data stream, the FDL-Macro switches back to HDLC-mode.

Once in BOM mode, if eight consecutive ones are not detected in 32 bits, a BOM header error will be declared.

Transmission of BOM data is done via the transparent mode of the signalling controller.

BOM Regular Mode

The following byte format is assumed (the left most bit is received first):

111111110xxxxxx0_B

The signalling controller uses the FF_H byte for synchronization, the next byte is stored in the receive FIFO (first bit received: LSB) if it starts and ends with a '0'. Bytes starting or ending with a '1' are not stored. If there are no 8 consecutive one's detected within 32 bits and the FDL-Macro is currently in the BOM mode, an 'Incorrect Synchronization Format' interrupt vector is generated. However, byte sampling is not stopped.

After detecting an HDLC flag, byte sampling is stopped, the receive status byte marking a BOM frame is stored in the receive FIFO and a 'Receive Message End' interrupt vector is generated.

Byte sampling may be stopped by deactivating the BOM receiver (RCR1.BRAC). In this case the receive status byte marking a BOM frame is added, a 'Receive Message End' interrupt vector is generated and HDLC mode is entered.

BOM Filter Mode

In BOM filter mode the received BOM data is validated and then filtered. If same valid BOM pattern is received for 7 out of 10 patterns, then BOM data is written to the receive FIFO along with the status byte indicating that filtered BOM data was received.

Filtered BOM mode will be exited if one of the following conditions occurs:

- 4 valid BOM patterns are consecutively received but none of these equals the BOM data received earlier.
- 4 times idle pattern is received.
- A HDLC flag is received.

4.8.4 S_a-bit Access

The MUNICH256FM supports the S_a-bit signaling of time-slot 0 of the T1/E1 signals in several ways. The access via registers RSAW and XSAW, capable of storing the information for a complete multiframe, and the most effective one is the access via the receive/transmit FIFOS of the integrated signaling controller.

Functional Description

The extended S_a -bit access gives the opportunity to transmit/receive a transparent bit stream as well as HDLC frames where the signaling controller automatically processes the HDLC protocol.

Data written to the transmit FIFO will subsequently be transmitted in the selected S_a -bit positions. Any combination of S_a -bits can be selected. After the data have been completely sent out an "all ones" or flags will be transmitted. The continuous transmission of a transparent bit stream, which is stored in the XFF.XFIFO, can be enabled.

The access to and from the FIFOs is supported by status and interrupts.

S_a -Bit Detection according to ETS 300233

Four consecutive received S_a -bits are checked on the by ETS 300233 defined S_a -bit combinations. The MUNICH256FM can be programmed to detect any bit combination on one S_a -bit out of S_{a4} through S_{a8} . Enabling of specific bit combination can be done via register RCR2.SASSM. A valid S_a -bit combination must occur three times in a row. The corresponding status in register RSAW4 will be set. Register RSAW4 is from type "Clear on Read". With any change of state of the selected S_a -bit combinations a 'SSM Data Valid' interrupt vector will be generated.

During the basic frame asynchronous state updating of register RSAW4 and interrupt vector generation is disabled. In CRC-4 multiframe format the detection of the S_a -bit combinations can be done either synchronous or asynchronous to the submultiframe. In synchronous detection mode updating of register RSAW4 is done in the multiframe synch. state. In asynchronous detection mode updating is independent to the multiframe synchronous state.

S_a -bit Error Indication Counters

The S_a -bit error indication counter CRC1 (16 bits) counts either the received bit sequence 0001_B or 0011_B or user programmable values in every submultiframe on a selectable S_a -bit. In the primary rate access digital section CRC errors are reported from the TE via S_{a6} . Incrementing is only possible in the multiframe synchronous state.

The S_a -bit error indication counter CRC2 (16 bits) counts either the received bit sequence 0010_B or 0011_B or user programmable values in every submultiframe on a selectable S_a -bit. In the primary rate access digital section CRC errors detected at T-reference points are reported via S_{a6} . Incrementing is only possible in the multiframe synchronous state.

4.8.5 Signalling Controller FIFO Operations

Access to the FIFO's of the signalling controllers is handled via registers RFF and XFF. FIFO status and commands are exchanged using the port status registers PSR and the

Functional Description

handshake register HND. Additional facility data link interrupt vectors inform system software about protocol and FIFO status.

Receive FIFO

In receive direction there are different interrupt indications associated with the reception of data:

- A 'Receive Pool Full' (RPF) interrupt vector is indicating that a data block can be read from the receive FIFO and the received message is not yet complete. It is generated, when the amount of data bytes has reached the programmed threshold.
- A 'Receive Message End' (RME) interrupt vector is indicating that the reception of one message is completed. After this interrupt system software has to read the PSR register in order to get the number of bytes stored in the receive FIFO. This number includes the status byte which is written into the receive FIFO as the last byte after the received frame. The status byte includes information about the CRC result, valid frame indication, abort sequence or data overflow. The format of the status byte is shown in the table below:

| | | | | |
|------------|---|------|-----------|---|
| 7 | 6 | 5 | 4 | 0 |
| SMODE(1:0) | | BRFO | STAT(4:0) | |

- SMODE** Receiver Status Mode
This bit indicates the type of data received.
- 10_B HDLC data
01_B BOM data
- BRFO** BOM Receive FIFO Overflow
- 0 No overflow
1 Receive FIFO overflow
- STAT** Receive FIFO Status
This bit field reports the status of the data stored in the receive FIFO.

| | HDLC mode | BOM MODE |
|--------------------|-----------------------|--|
| 00000 _B | Valid HDLC Frame | BOM Filtered data declared |
| 00001 _B | Receive Data Overflow | BOM data available |
| 00010 _B | Receive Abort | BOM End |
| 00011 _B | Not Octet | BOM filtered data undeclared |
| 00100 _B | CRC Error | BOM header error (ISF, incorrect synchronization format) |
| 00101 _B | Channel Off | |

Functional Description

After the received data has been read from the FIFO, the receive FIFO can be released by the CPU by issuing a 'Receive Message Complete' (HND.RMC) command. The CPU has to process a 'Receive Pool Full' interrupt vector and issue the 'Receive Message Complete' command before the second page of the FIFO becomes full. Otherwise a 'Receive Data Overflow' condition will occur. This time is dependent on the threshold programmed (smaller threshold results in shorter time).

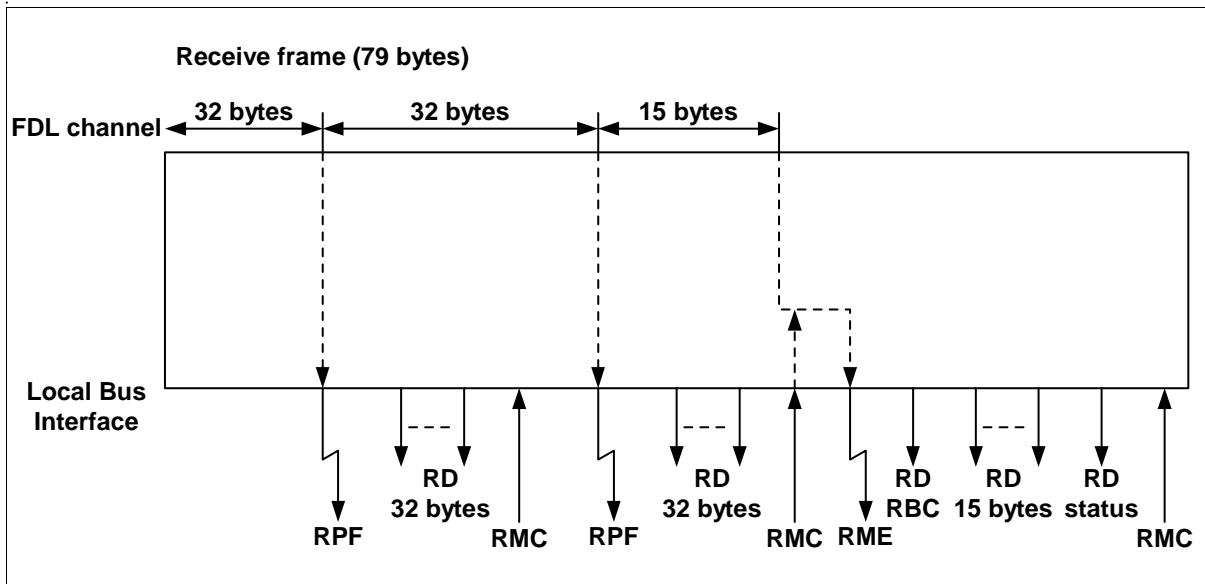


Figure 4-12 Interrupt Driven Reception Sequence Example

Transmit FIFO

In the transmit direction after checking the transmit FIFO status by polling the transmit FIFO write enable bit (PSR.XFW) or after a 'Transmit Pool Ready' (XPR) interrupt vector, up to 32 bytes may be written to the transmit FIFO (bit field XFF.XFIFO) by the CPU. Transmission of a frame can be started by issuing a 'Transmit Transparent Frame' (XTF) or 'Transmit HDLC Frame' (XHF) command via register HND. If the transmit command does not include a 'Transmit Message End' indication (HND.XME), the signalling controller will repeatedly request for the next data block by means of a XPR interrupt vector as soon as the transmit FIFO becomes free. This process will be repeated until the local CPU writes the last bytes to the transmit FIFO. The end of message is then indicated per HND.XME command, after which frame transmission is finished correctly by appending the CRC and closing flag sequence. Consecutive frames may share a flag (enabled via bit XCR1.SF) or may be transmitted as back-to-back frames, if service of transmit FIFO is quick enough. In case that no more data is available in the transmit FIFO prior to the arrival of HND.XME, the transmission of the frame is terminated with an abort sequence and the CPU is notified via a 'Transmit Data Underrun' interrupt vector (XDU). The frame may also be aborted per software by setting the XAB bit in the handshake register HND.

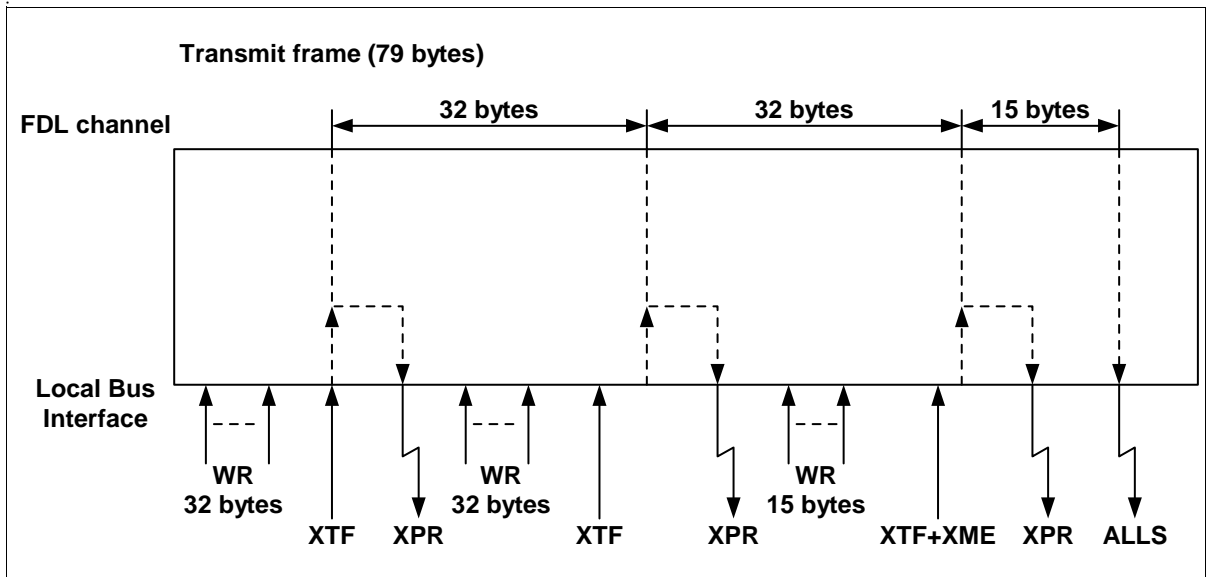


Figure 4-13 Interrupt Driven Transmit Sequence Example

Note: Transmit FIFO is 16 bit wide. In the given example writing 32 bytes requires 16 write accesses. Writing 15 byte requires 8 accesses.

4.9 M12 Multiplexer/Demultiplexer and DS2 framer

The M12 multiplexer and the DS2 framer can be operated in two modes:

- M12 multiplex format according to ANSI T1.107
- ITU-T G.747 format

4.9.1 M12 multiplex format

The framing structure of the M12 signal is shown in **Table 4-11**. A DS2 multiframe consists of four subframes. Each subframe combines 6 blocks with 49 bits each. The first bit of each block contains an overhead (OH) bit and 48 information bits. The 48 information bits are divided into four time slots of 12 bits each. The first time slot is assigned to the 1st tributary DS1 signal, the second time slot is assigned to the 2nd tributary DS1 signal and so forth.

Table 4-11 M12 multiplex format

| | Subframe | Block 1 through 6 of a subframe | | | | | |
|--------------------|----------|---------------------------------|----------------------|---------------------|----------------------|----------------------|---------------------|
| | | 1 | 2 | 3 | 4 | 5 | 6 |
| DS2- Multiframe | 1 | 0 _M [48] | C ₁₁ [48] | F ₀ [48] | C ₁₂ [48] | C ₁₃ [48] | F ₁ [48] |
| | 2 | 1 _M [48] | C ₂₁ [48] | F ₀ [48] | C ₂₂ [48] | C ₂₃ [48] | F ₁ [48] |
| | 3 | 1 _M [48] | C ₃₁ [48] | F ₀ [48] | C ₃₂ [48] | C ₃₃ [48] | F ₁ [48] |
| | 4 | X [48] | C ₄₁ [48] | F ₀ [48] | C ₄₂ [48] | C ₄₃ [48] | F ₁ [48] |

F₀, F₁

F₀ and F₁ form the frame alignment pattern. Each DS2 frame consists of eight F-bits, two per subframe in block 3 and 6. F₀ and F₁ form the pattern '01'. This pattern is repeated in every subframe.

X

This bit is the forth bit of the multiframe alignment signal and can be set to either '0' or '1'. It is accessible via an internal register.

M₀, M₁, M_X

M₀ and M₁ and M_X form the multiframe alignment signal. Each subframe consists of four M-bits and they are located in bit 0 of each subframe. The multiframe alignment signal is '011-'.

C₁₁..C₄₃

The C-bits control the bit stuffing procedure of the multiplexed DS1 signals.

[48]

These bits represent a data block, which consists of 48 bits. [48] consists of four time slots of 12 bit and each time slot is assigned to one of four participating DS1 signals.

4.9.1.1 Synchronization Procedure

The integrated DS2 framer searches for the frame alignment pattern '01' and the multiframe alignment pattern in each of the seven DS2 frames which are contained in a DS3 signal. Frame alignment is declared, when the DS2 framer has found the basic frame alignment pattern (F-bit) and the multiframe alignment pattern (M-bit).

Loss of frame is declared, when 2 out of 4 or 3 out of 5 incorrect F-bits are found or when one or more incorrect M-bits are found in 3 out of 4 subframes.

4.9.1.2 Multiplexer/Demultiplexer

Demultiplexer

The demultiplexer extracts four DS1 signals out of each DS2 signal. If two out of three bits of C_{i1} , C_{i2} , C_{i3} are set to '1' the first information bit in the i^{th} subframe and the 6th block which is assigned to the i^{th} DS1 signal is discarded.

The demultiplexer performs inversion of the 2nd and 4th tributary DS1 signal.

Multiplexer

The multiplexer combines four DS1 signals to form a DS2 signal. Stuffing bits are inserted and the C_{i1} -, C_{i2} -, C_{i3} -bits, which are assigned to the i^{th} DS1 signal, are set to '1' in case that not enough data is available.

The 2nd and 4th DS1 signal are automatically inverted in transmit direction.

4.9.1.3 Loopback Control

Detection

Loopback requests encoded in the C-bits of the DS2 signal are flagged when they are repeated for at least five DS2 multiframes. Loops must be initiated by an external microprocessor.

Generation

A loopback request, which is transmitted in lieu of the C-bits, can be placed in each DS2 signal.

4.9.1.4 Alarm Indication Signal

Detection

AIS is declared, when the AIS condition (the received DS2 data stream contains an all '1' signal with less than 3/9 zeros within 3156 bits while the DS2 framer is out of frame) is present within a time interval that is determined by register D2RAP.

Generation

The alarm indication signal is an all '1' unframed signal and will be transmitted if enabled.

4.9.2 ITU-T G.747 format

The multiplexing frame structure is shown in **Table 4-12**.

Table 4-12 ITU-T G.747 format

| | Set | Content | Bit |
|-------------------------|-----|--|-----------|
| ITU-T G.747 Frame | I | Frame Alignment Signal 111010000 | 1 to 9 |
| | | Bits from tributaries | 10 to 168 |
| | II | Alarm indication to the remote multiplex equipment | 1 |
| | | Parity Bit | 2 |
| | | Reserved | 3 |
| | | Bits from tributaries | 4 to 168 |
| | III | Justification control bits C_{j1} | 1 to 3 |
| | | Bits from tributaries | 4 to 168 |
| | IV | Justification control bits C_{j2} | 1 to 3 |
| | | Bits from tributaries | 4 to 168 |
| | V | Justification control bits C_{j3} | 1 to 3 |
| | | Bits from tributaries available for justification | 4 to 6 |
| | | Bits from tributaries | 7 to 168 |

4.9.2.1 Synchronization Procedure

The integrated framer searches for the frame alignment pattern '111010000' in each of the seven frames which are contained in a DS3 signal. Frame alignment is declared, when the framer has found three consecutive correct frame alignment signals. If the frame alignment signal has been received incorrectly in one of the following frames after the receiver found the first correct frame alignment signal a new search is started.

Loss of frame is declared, when four consecutive frame alignment signals have been received incorrectly.

4.9.2.2 Multiplexer/Demultiplexer

Demultiplexer

The demultiplexer extracts three E1 signals from each 6.312 MHz signal. If two out of three bits of C_{j1} , C_{j2} , C_{j3} are set to '1' the available justification bit of the j^{th} E1 signal is discarded.

Multiplexer

The multiplexer combines three E1 signals to form a DS2 signal. Stuffing bits are inserted and the C_{j1} -, C_{j2} -, C_{j3} -bits, which are assigned to the j^{th} E1 signal, are set to '1' in case that not enough data is available.

4.9.2.3 Parity Bit

Detection

The receiver optionally calculates the parity of all tributary bits and compares this value with the received parity bit. Differences are counted in the parity error counter.

Generation

The parity bit is automatically calculated according to ITU-T G.747 or programmable to a fixed value under microprocessor control.

4.9.2.4 Remote Alarm Indication

Detection

Remote alarm is reported when bit 1 of set II changes and when the change persists for at least three multiframes.

Generation

Remote alarm is transmitted in bit 2 of "set II" and can be inserted under microprocessor control.

4.9.2.5 Alarm Indication Signal

Detection

AIS is declared, when the AIS condition (the received DS2 data stream contains an all '1' signal with less than 5/9 zeros within two consecutive multiframes while the DS2 framer is out of frame) is present within a time interval that is determined by register D2RAP.

Generation

The alarm indication signal is an all '1' unframed signal and will be transmitted if enabled.

4.10 M23 multiplexer and DS3 framer

The M23 multiplexer and the DS3 framer can be operated in three modes:

- M23 multiplex format
- C-bit parity format with modified M23 multiplex operation
- C-bit parity format with non-M23 multiplex operation (Full payload rate format)

4.10.1 M23 multiplex format

The framing structure of the M23 multiplex signal is shown in **Table 4-13**. Each DS3 multiframe consists of 7 subframes and each subframe of eight blocks. One block consists of 85 bits, where the first bit is the overhead (OH) bit and the remaining 84 bits are the information bits. The 84 information bits are divided into seven time slots of 12 bits each. The first time slot is assigned to the 1st tributary DS2 signal, the second time slot is assigned to the 2nd tributary DS2 signal and so forth.

Table 4-13 M23 multiplex format

| | Sub-frame | Block 1 through 8 of a subframe | | | | | | | |
|-----------------|-----------|---------------------------------|---------------------|----------------------|---------------------|----------------------|---------------------|----------------------|---------------------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| DS3-Multi-frame | 1 | X [84] | F ₁ [84] | C ₁₁ [84] | F ₀ [84] | C ₁₂ [84] | F ₀ [84] | C ₁₃ [84] | F ₁ [84] |
| | 2 | X [84] | F ₁ [84] | C ₂₁ [84] | F ₀ [84] | C ₂₂ [84] | F ₀ [84] | C ₂₃ [84] | F ₁ [84] |
| | 3 | P [84] | F ₁ [84] | C ₃₁ [84] | F ₀ [84] | C ₃₂ [84] | F ₀ [84] | C ₃₃ [84] | F ₁ [84] |
| | 4 | P [84] | F ₁ [84] | C ₄₁ [84] | F ₀ [84] | C ₄₂ [84] | F ₀ [84] | C ₄₃ [84] | F ₁ [84] |
| | 5 | M ₀ [84] | F ₁ [84] | C ₅₁ [84] | F ₀ [84] | C ₅₂ [84] | F ₀ [84] | C ₅₃ [84] | F ₁ [84] |
| | 6 | M ₁ [84] | F ₁ [84] | C ₆₁ [84] | F ₀ [84] | C ₆₂ [84] | F ₀ [84] | C ₆₃ [84] | F ₁ [84] |
| | 7 | M ₀ [84] | F ₁ [84] | C ₇₁ [84] | F ₀ [84] | C ₇₂ [84] | F ₀ [84] | C ₇₃ [84] | F ₁ [84] |

F₀, F₁

F₀ and F₁ form the frame alignment pattern. Each DS3 frame consists of 28 F-bits, four per subframe in block 2, 4, 6 and 8. F₀ and F₁ form the pattern '1001'. This pattern is repeated in every subframe.

M₀, M

M₀ and M₁ form the multiframe alignment signal. The M-bit is contained in the OH-bit of the first block in subframe 5,6 and 7. The multiframe alignment signal is '010'.

C₁₁..C₇₃

The C-bits control the bit stuffing procedure of the multiplexed DS2 signals.

P

The P-bits contain parity information and are calculated as even parity on all information bits of the previous DS3 frame. Both P-bits are identical.

X

The X-bits are used for transmission of asynchronous in-service messages. Both X-bits must be identical and may not change more than once every second.

[84]

These bits represent a data block, which consists of 84 bits.

[84] consists of seven time slots with 12 bits each and they are assigned to one of the seven participating DS2 signals.

4.10.1.1 Synchronization Procedure

The integrated DS3 framer searches for the frame alignment pattern '1001' and when found for the multiframe alignment pattern in each of the seven DS3 subframes. When the multiframe alignment pattern is found in three consecutive DS3 frames while frame alignment is still valid frame alignment is declared. The P-bits and the X-bits are ignored during synchronization.

Loss of frame is declared, when 3 out of 8 or 3 out of 16 incorrect F-bits are found or when one or more incorrect M-bits are found in 3 out of 4 subframes.

4.10.1.2 Multiplexer/Demultiplexer

Demultiplexer

The demultiplexer extracts seven DS2 signals from the incoming DS3 signal. If two or three bits out of C_{i1} , C_{i2} , C_{i3} are set to '1' the first bit following the F_1 bit in the i^{th} subframe which is assigned to the i^{th} DS2 signal is discarded.

Multiplexer

The multiplexer combines seven DS2 signals to form a DS3 signal. If not sufficient data is available for a DS2 signal, it automatically inserts a stuffing bit and sets the bits C_{i1} , C_{i2} , C_{i3} assigned to the i^{th} DS2 signal to '1'.

4.10.1.3 X-bit

The MUNICH256FM provides access to the X-bit of each tributary via an internal registers. Data written to the X-bit register is copied to an internal shadow register which is then locked for one second after each write access.

4.10.1.4 Alarm Indication Signal, Idle Signal

Detection

Alarm indication signal or Idle signal is declared, when the selected signal format was received with less than 8/15 bit errors (selectable via bit D3RAP.AIS) for at least one multiframe. The alarm indication signal can be selected as:

- Unframed all '1's
- Framed '1010' sequence, starting with a binary '1' after each OH-bit. C-bits are set to '0'. X-bit can be checked as '1' or X-bit check can be disabled.

The idle signal is a

- Framed '1100' sequence, starting with a binary '11' after each OH-bit. C-bits are set to '0' in M-subframe 3. X-bit can be checked as '1' or X-bit check can be disabled.

Generation

The alarm indication signal or idle signal will be generated according to the selected signal format. X-bit needs to be set separately to '1'.

4.10.1.5 Loss of Signal

Detection

Loss of signal is declared, when the incoming data stream contains more than 1022 consecutive '0's.

Recovery

Loss of signal is removed, when two or more ones are detected in the incoming data stream.

4.10.1.6 Performance Monitor

The following conditions are counted:

- Line code violations
- Excessive zeroes
- P-bit errors, CP-bit errors
- Framing bit errors
- Multiframe bit errors
- Far end block errors

4.10.2 C-bit parity format

The framing structure of the C-bit parity format is shown in **Table 4-13**. The assignment of the information bits [84] is identical to the M23 multiplex format, but the function of the C-bits is redefined for path maintenance and data link channels.

Table 4-14 C-bit parity format

| | Sub-frame | Block 1 through 8 of a subframe | | | | | | | |
|-----------------|-----------|---------------------------------|---------------------|----------------------|---------------------|----------------------|---------------------|----------------------|---------------------|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| DS3-Multi-frame | 1 | X [84] | F ₁ [84] | AIC [84] | F ₀ [84] | N _r [84] | F ₀ [84] | FEAC [84] | F ₁ [84] |
| | 2 | X [84] | F ₁ [84] | DL [84] | F ₀ [84] | DL [84] | F ₀ [84] | DL [84] | F ₁ [84] |
| | 3 | P [84] | F ₁ [84] | CP [84] | F ₀ [84] | CP [84] | F ₀ [84] | CP [84] | F ₁ [84] |
| | 4 | P [84] | F ₁ [84] | FEBE [84] | F ₀ [84] | FEBE [84] | F ₀ [84] | FEBE [84] | F ₁ [84] |
| | 5 | M ₀ [84] | F ₁ [84] | DL _t [84] | F ₀ [84] | DL _t [84] | F ₀ [84] | DL _t [84] | F ₁ [84] |
| | 6 | M ₁ [84] | F ₁ [84] | DL [84] | F ₀ [84] | DL [84] | F ₀ [84] | DL [84] | F ₁ [84] |
| | 7 | M ₀ [84] | F ₁ [84] | DL [84] | F ₀ [84] | DL [84] | F ₀ [84] | DL [84] | F ₁ [84] |

F₀, F₁

F₀ and F₁ form the frame alignment pattern. Each DS3 frame consists of 28 F-bits, four per subframe in block 2, 4, 6 and 8. F₀ and F₁ form the pattern '1001'. This pattern is repeated in every subframe.

M₀, M₁

M₀ and M₁ form the multiframe alignment signal. The M-bit is contained in the OH-bit of the first block in subframe 5,6 and 7. The multiframe alignment signal is '010'.

N_r

Reserved. Set to '1' in transmit direction.

AIC

Application Identification Channel.

DL_t

The terminal-to-terminal path maintenance data link uses the HDLC protocol. Access to the DL_t bits is possible via the DS3 transmit and receive FIFO.

DL

Reserved. Set to '1' in transmit direction.

FEAC

The alarm or status information of a far end terminal is sent back over the far end and control channel. This bit also contains DS3 or DS1 line loopback requests. Messages are sent in bit oriented mode. Message codes can be accessed via an internal register.

FEBE

The far end block error bits indicate a CP-bit parity error or a framing error. They are used to

Functional Description

monitor the performance of a DS3 signal. Upon detection of either error in the incoming data stream the FEBE-bits are set automatically to '000' in the outgoing direction. Received far end block errors are counted.

CP

The CP-bits are used to carry path parity information and are set to the same value as the P-bits. In receive direction the CP-bits are checked against the calculated parity and differences are counted.

P

The P-bits contain parity information and are automatically calculated as even parity on all information bits of the previous DS3 frame.

X

The X-bits are used for transmission of asynchronous in-service messages. Both X-bits must be identical and may not change more than once every second. Access to the X-bits is possible via a register.

[84]

These bits represent a data block, which consists of 84 bits. [84] consists of seven time slots with 12 bits each and they are assigned to one of the seven participating DS2 signals.

4.10.2.1 Synchronization Procedure

The integrated DS3 framer searches for the frame alignment pattern '1001' and when found for the multiframe alignment pattern in each of the seven DS3 subframes. Frame alignment is declared when the multiframe alignment pattern is found in three consecutive DS3 frames. The P-bits and the X-bits are ignored during synchronization.

Loss of frame is declared, when 3 out of 8 or 3 out of 16 incorrect F-bits are found or when one or more incorrect M-bits are found in 3 out of 4 subframes.

4.10.2.2 Multiplexer/Demultiplexer

Demultiplexer

The demultiplexer extracts seven DS2 signals from the incoming DS3 signal. Since the DS3 signal is always stuffed the stuffing bit assigned to each DS2 signal is discarded.

Multiplexer

The multiplexer combines seven DS2 signals to form a DS3 signal and automatically inserts a stuffing bit for each DS2 signal.

4.10.2.3 X-bit

The MUNICH256FM provides access to the X-bits via internal registers.

4.10.2.4 Far End Alarm and Control Channel

The far end alarm and control channel is accessible via the signalling controller in BOM mode.

4.10.2.5 Path Maintenance Data Link Channel

The path maintenance data link channel is accessible via the signalling controller in HDLC mode.

4.10.2.6 Loopback Control

Detection

Loopback requests are encoded in the messages of the far end alarm and control channel. The microprocessor has access to the messages as described in **Chapter 4.10.2.4**.

Generation

A loopback request can be initiated via the far end alarm and control channel.

4.10.2.7 Alarm Indication Signal, Idle Signal

Detection

Alarm indication signal or Idle signal is declared, when the selected signal format was received with less than 8/15 bit errors (selectable via bit D3RAP.AIS) for at least one multiframe. The alarm indication signal can be selected as:

- Unframed all '1's
- Framed '1010' sequence, starting with a binary '1' after each OH-bit. C-bits are set to '0'. X-bit can be checked as '1' or X-bit check can be disabled.

The idle signal is a

- Framed '1100' sequence, starting with a binary '11' after each OH-bit. C-bits are set to '0' in M-subframe 3. X-bit can be checked as '1' or X-bit check can be disabled.

Generation

The alarm indication signal or idle signal will be generated according to the selected signal format. X-bit needs to be set separately to '1'.

4.10.2.8 Loss of Signal

Detection

Loss of signal is declared, when the incoming data stream contains more than 1022 consecutive '0's.

Recovery

Loss of signal is removed, when two or more ones are detected in the incoming data stream.

4.10.2.9 Performance Monitor

The following conditions are counted:

- Line code violations
- Excessive zeroes
- P-bit errors, CP-bit errors
- Framing bit errors
- Multiframe bit errors
- Far end block errors

4.10.3 Full Payload Rate Format

In full payload rate format the DS3 multiframe structure can be selected according to the M13 multiplex structure or the C-bit parity structure. In either case the data blocks [84] carry one continuous data stream which is provided via the tributary interface one.

Multiplexing/Demultiplexing of the data block [84] does NOT apply.

4.11 Test Unit

The test unit of the MUNICH256FM incorporates a test pattern generator and a test pattern synchronizer which can be attached to different test points as shown in **Figure 4-14**. Controlled by a small set of registers it can generate and synchronize to polynomial pseudorandom test patterns or repetitive fixed length test patterns.

Test patterns can be generated in the following modes:

- Framed DS3
- Unframed DS2
- Framed DS2
- Unframed DS1/E1

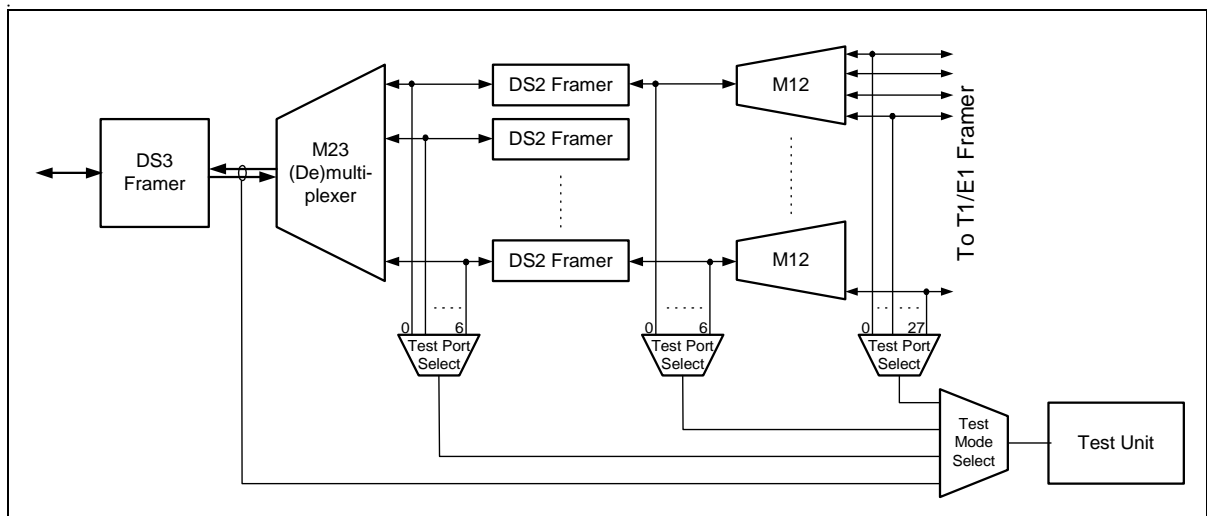


Figure 4-14 Test Unit Access Points

In pseudorandom test mode the receiver tries to achieve synchronization to a test pattern which satisfies the programmed receiver polynomial. In fixed pattern mode it synchronizes to a repetitive pattern with a programmable length. An all '1' pattern or an all '0' pattern, which satisfies this condition, is flagged. Measurement intervals as well as receiver synchronization can be controlled by the user. When a test is finished an interrupt is generated and the bit count and the bit error count are readable.

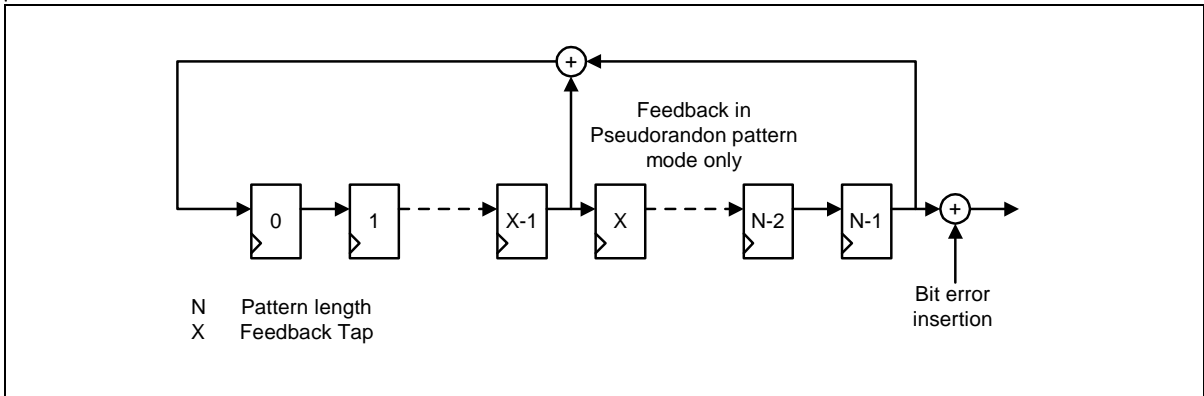


Figure 4-15 Pattern Generator

Bit Error Insertion

The test unit provides the optional capability to insert bit errors in the range of 10^{-7} (1 error in 10.000.000 bits) up to 10^{-1} bit errors (1 error in 10 bits).

4.12 Mailbox

The MUNICH256FM contains a mailbox to allow communication between two intelligent peripherals connected to the PCI bus and the local microprocessor bus. The mailbox is organized in two pages of eight registers. The first page is used to store information from the PCI side and to read the information from the local microprocessor side. The second page is used for the opposite direction, from the local microprocessor side to the PCI side. Each page consists of one status register and seven data registers.

The mailbox provides a 'doorbell' capability. In this case an interrupt vector can be generated to inform the addressed intelligent peripheral that new information has been stored in the mailbox. This interrupt vector will be generated on write accesses to the status register of the selected page.

As an **example**, consider when the PCI host system wants to transfer data to an intelligent peripheral. First it loads data into the mailbox data registers MBP2E1 through MBP2E7, and then writes a status information to the mailbox status register MBP2E0. This last action causes an interrupt vector to be written to the interrupt FIFO which is connected to the local bus. The presence of an interrupt vector results in assertion of pin $\overline{\text{LINT}}$. The intelligent peripheral recognizes the interrupt pin asserted and reads the interrupt vector out of the interrupt FIFO (which results in deassertion of pin $\overline{\text{LINT}}$), and then reads data from the mailbox data registers.

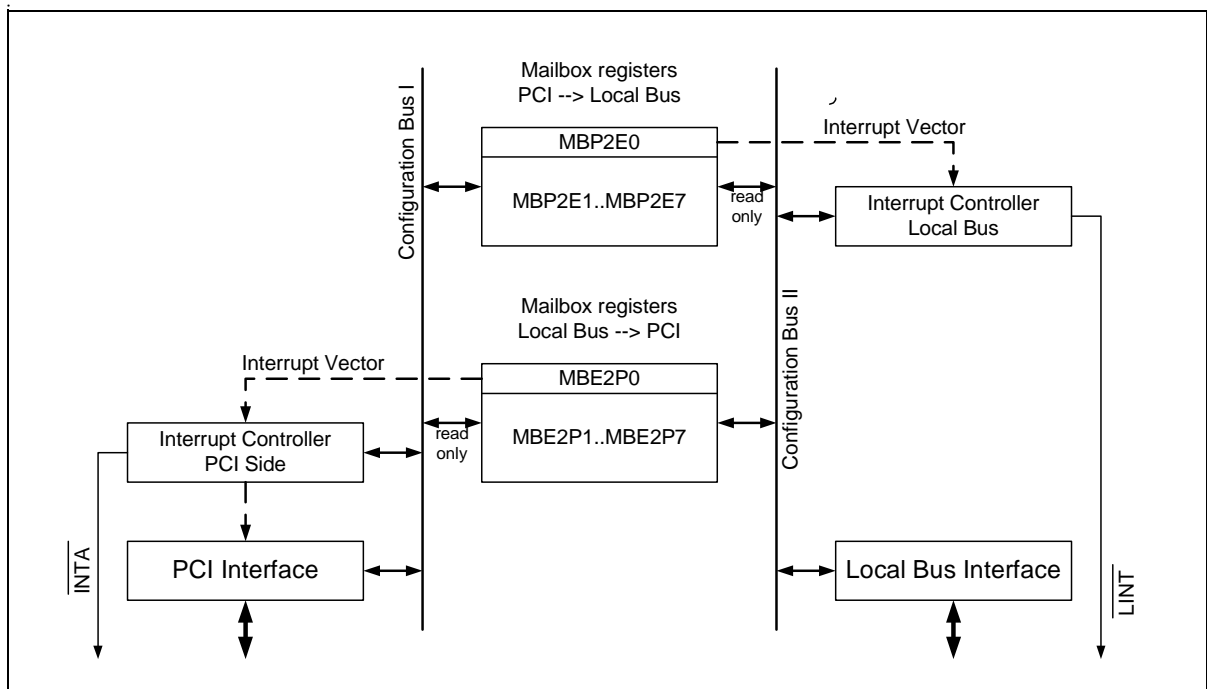


Figure 4-16 Mailbox Structure

Alternately, consider when an intelligent peripheral connected to the local bus wants to transfer data to the PCI host system. First it loads data into the mailbox data registers

Functional Description

MBE2P1 through MBE2P7 and then it writes status information to the mailbox status register MBE2P0. This causes a system interrupt vector to be written to the PCI host system, indicating that valid data is contained in the mailbox data registers.

This interrupt vector will be written to the interrupt queue specified in CONF1.SYSQ and together with this the pin $\overline{\text{INTA}}$ will be asserted. The processor sees the interrupt pin asserted, reads the register GISTA in order to determine the interrupt queue, and then writes a '1' to the interrupt status acknowledge register GIACK to clear the interrupt. Next, it reads the interrupt vector which contains a copy of the mailbox status register and then reads the mailbox data registers.

4.13 Interrupt Controller

Since the MUNICH256FM is divided into the basic functions mailbox, layer one functions (T1/E1 framer, facility data link, M13 multiplexer and DS2/DS3 framer) and layer two protocol functions (HDLC, PPP, TMA), the same partitioning is used for the interrupt handling.

All layer two interrupts (channel, port, system and command interrupts) are handled via an internal interrupt controller which forwards those interrupts to external interrupt queues. This interrupt controller is connected to the PCI interrupt pin \overline{INTA} .

Mailbox interrupts and layer one interrupts are handled via an internal interrupt FIFO which is connected to the local bus interrupt pin \overline{LINT} (normal operation). Additionally the interrupts stored in the internal interrupt FIFO can be notified via the PCI interrupt pin \overline{INTA} .

The MUNICH256FM also provides the capability to bridge the local bus interrupt \overline{LINT} to the PCI bus.

4.13.1 Layer Two interrupts

All channel interrupts, port interrupts and system interrupts are written in form of interrupt vectors to interrupt queues.

Each interrupt vector has an interrupt source. An interrupt source is either a channel, the port handler or certain device functions (system interrupts). After reset no interrupt vector is generated since port and system interrupts are masked and channels are in their idle state.

Each interrupt source forwards its interrupt vector to the interrupt controller, together with the information in which interrupt queue the vector should be forwarded. The interrupt controller moves the interrupt vector to the selected interrupt queue. Channel interrupts can optionally be forwarded to a dedicated high priority interrupt queue (interrupt queue seven). A programmable interrupt queue high priority mask determines channel interrupts, which shall be forwarded into the high priority interrupt queue instead of queueing them in the selected interrupt queue. This function is available for each interrupt queue and allows to queue important interrupt conditions in the high priority queue.

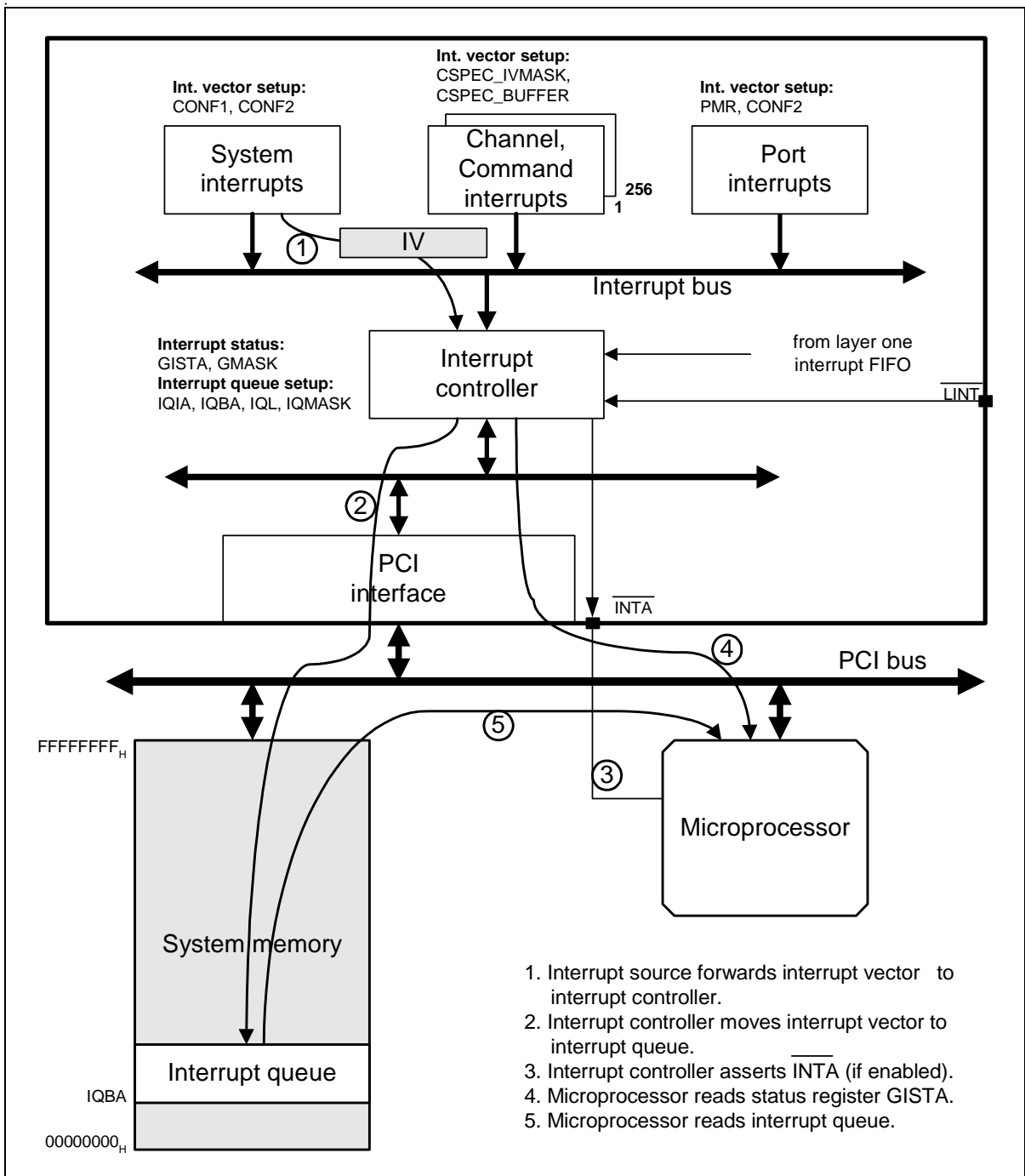


Figure 4-17 Layer Two Interrupts (Channel, command, port and system interrupts)

As soon as the interrupt controller has written an interrupt vector to one of the nine interrupt queues the PCI interrupt pin $\overline{\text{INTA}}$ is asserted. The global interrupt status register indicates in which interrupt queue the interrupt vector can be found. Each of the

Functional Description

nine interrupt queues can be masked. In this case the interrupt pin \overline{INTA} is not asserted, but the interrupt vector is still written into the assigned interrupt queue.

An interrupt queues is a reserved memory locations in system memory. The MUNICH256FM supports up to eight interrupt queues which are organized in form of ring buffers with a programmable start address and a programmable size per interrupt queue. Additionally there is one fixed sized command interrupt queue where command interrupts are stored. The size of this queue is two times 256 DWORDs (**Figure 4-18**).

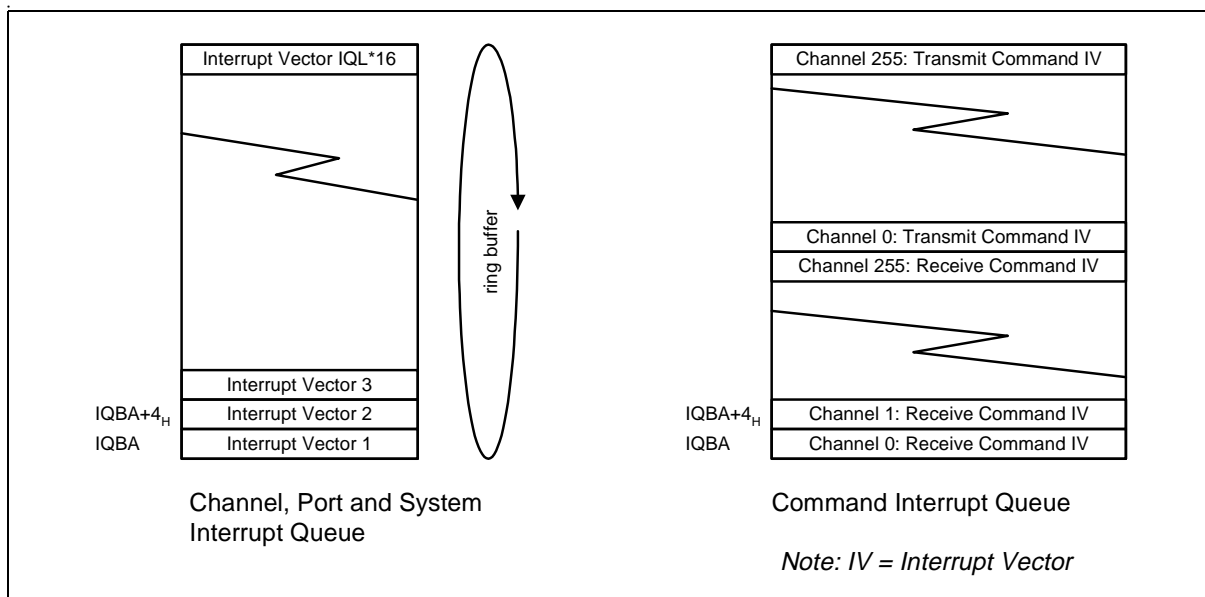


Figure 4-18 Interrupt Queue Structure in System Memory

4.13.1.1 General Interrupt Vector Structure

Each interrupt vector is 32 bit wide and contains several subfields, which indicate the interrupt group and depend on the interrupt group the interrupt information. Bit 31 of the interrupt vector is generally set to '1' by the MUNICH256FM and allows the system CPU to clear the bit in order to mark processed interrupts.

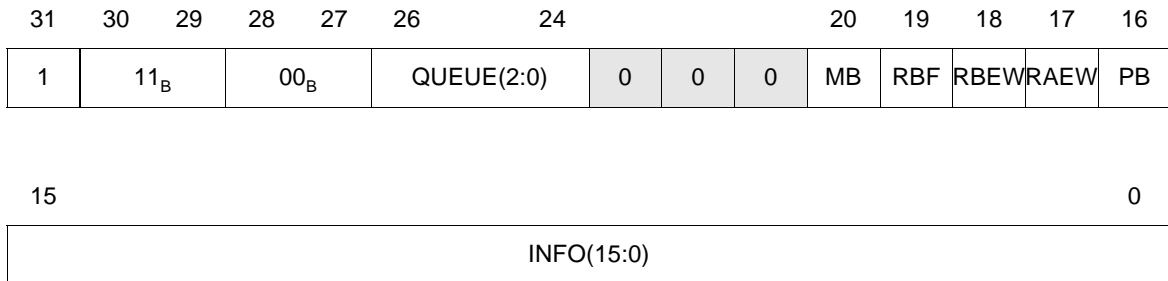
Table 4-15 Interrupt Vector Structure

| | | | | | | | | |
|-----------|-----------|------------|------------|-----------|----|----|----|----|
| 31 | 30 | 29 | 28 | 27 | 26 | 24 | 23 | 16 |
| 1 | TYPE(1:0) | STYPE(1:0) | QUEUE(2:0) | INT(23:0) | | | | |
| 15 | | | | | | | | 0 |
| INT(23:0) | | | | | | | | |

Functional Description

| | |
|-------|---|
| TYPE | <p>Interrupt type</p> <p>The interrupt vectors are divided into four basic groups, where TYPE determines the interrupt group. A further classification of interrupts is done with the subtype indication.</p> <p>00_B Command interrupts</p> <p>01_B Channel interrupts</p> <p>10_B Port interrupts</p> <p>11_B System interrupts</p> |
| STYPE | <p>Interrupt subtype</p> <p>A specific interrupt type is divided into several subtypes. In general STYPE(1) indicates the data path (transmit, receive) generating the interrupt.</p> |
| QUEUE | <p>Interrupt queue</p> <p>The interrupt vectors are written into 9 external interrupt queues located in the shared memory. Corresponding to these 9 queues are 9 interrupt queue start addresses and 8 interrupt queue length registers, since the interrupt queue 8 has a fixed length of 2 x 256).</p> |
| INT | <p>Interrupt Information</p> <p>INT itself contains the interrupt information. The meaning of INT is dependent on TYPE and STYPE indication.</p> |

4.13.1.2 System Interrupts



- MB** Mailbox
The 'Mailbox' interrupt vector is generated, in case that the local microprocessor has written data to the mailbox status register MBE2P0. The bit field INFO contains a copy of MBE2P0.
- RBAF** Receive Buffer Access Failed
The 'Receive Buffer Access Failed' interrupt vector is generated, when the protocol machine discarded packets due to permanent inaccessibility of the receive buffer. This interrupt is issued as soon as the programmable threshold stored in register RBAFT is reached. The actual value of discarded packets is stored in register RBAFC.
- RBEW** Receive Buffer Queue Early Warning
The 'Receive Buffer Queue Early Warning' interrupt vector is generated, when the receive buffer data threshold has been exceeded (RBTH.RBTH). This interrupt can be masked via bit CONF1.RBIM.
- RAEW** Receive Buffer Action Queue Early Warning
The 'Receive Buffer Action Queue Early Warning' interrupt vector is generated, when the receive data action queue threshold (RBTH.RBAQTH) has been exceeded. The receive buffer action queue stores all requests of the receive buffer to forward data packets to system memory. This interrupt vector can be masked via bit CONF1.RBIM.
- PB** PCI Access Error
The 'PCI Access Error' interrupt vector is generated, when system software tries to read/write internal registers with accesses that do not enable all byte lanes, e.g. the access is not a full 32 bit access. The bit field INFO contains the register address which was tried to access.
- INFO** Contains additional interrupt information data according to the bit, which is set: See specific interrupt for details.

4.13.1.3 Port Interrupts

Port interrupt vectors indicate the synchronous or asynchronous state of a port. Immediately after enabling both, the port and the port interrupts, port interrupts are generated indicating the synchronous or asynchronous state of a port. After this initial interrupt vector generation, further interrupts are written only when the state of a port changes from synchronous state to asynchronous state or vice versa. Port interrupts are enabled by resetting the corresponding mask bit in register PMR.

Transmit interrupts

| | | | | | | | | | | | | | | | | | | | | | | | | |
|----|-----------------|----|-----------------|----|------------|---|---|---|---|---|----|-----------|---|---|---|---|-----|------|--|--|--|--|--|---|
| 31 | 30 | 29 | 28 | 27 | 26 | | | | | | 24 | | | | | | 17 | 16 | | | | | | |
| 1 | 10 _B | | 10 _B | | QUEUE(2:0) | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SYN | ASYN | | | | | | |
| | | | | | | | | | | | 15 | | | | | | 5 | 4 | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PORT(4:0) | | | | | | | | | | | | |

PORT Port Number
This bit field identifies the port for which the information in the interrupt vector is valid.

SYN Synchronization achieved
Port has changed from asynchronous state to synchronous state. This interrupt is available for ports configured in T1 or E1 mode. In unchannelized mode there is no synchronous state.

A transmit port changes to the synchronous state, if common transmit frame synchronization is enabled and the number of bits between two synchronization pulses is equal to the number of frame bits of the selected mode or is equal to a multiple of that number. The first CTFS pulse after a port is enabled causes the transmitter to change to the synchronous state.

In case the common transmit frame synchronization is disabled, i.e. the looped timing bit or the CTFS disable bit of a port is set in PMR, the initial asynchronous state will not be left.

ASYN Asynchronous State
The transmitter generates an 'Asynchronous State' interrupt vector if a port has changed from synchronous to asynchronous state. This interrupt is available for ports configured in T1 or E1 mode. In

Functional Description

unchannelized mode there is no asynchronous state. In general a port is in asynchronous state when a port is disabled.

A transmit port changes to the asynchronous mode if the number of bits between two synchronization pulses is not equal to a multiple of the number of frame bits of the selected mode

Receive Interrupts

| | | | | | | | | | | | | | |
|----|-----------------|-----------------|------------|----|----|----|---|---|---|---|-----------|-----|------|
| 31 | 30 | 29 | 28 | 27 | 26 | 24 | | | | | 17 | 16 | |
| 1 | 10 _B | 00 _B | QUEUE(2:0) | | | 0 | 0 | 0 | 0 | 0 | 0 | SYN | ASYN |
| 15 | | | | | | | | | | | 4 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | PORT(4:0) | | |

PORT

Port Number

This bit field identifies the port for which the information in the interrupt vector is valid.

SYN

Synchronization achieved

Port has changed from asynchronous state to synchronous state. This interrupt is available for ports configured in T1 or E1 mode. In unchannelized mode there is no synchronous state.

A receive port changes to the synchronous state, if the number of bits between two synchronization pulses generated by the port related framer is exactly equal to the number of frame bits of the selected mode. The first framer pulse after a port is enabled causes the receive port to change to the synchronous state.

ASYN

Asynchronous state

Port has changed from synchronous to asynchronous state. This interrupt is available for ports configured in T1 or E1 mode. In unchannelized mode there is no asynchronous state. In general a port is in asynchronous state when a port is disabled.

A receive port changes to the asynchronous state if the number of bits between two framer synchronization pulses is not equal to the number of frame bits of the selected mode. The synchronization pulses are generated internally by the T1/E1 framer.

4.13.1.4 Channel Interrupts

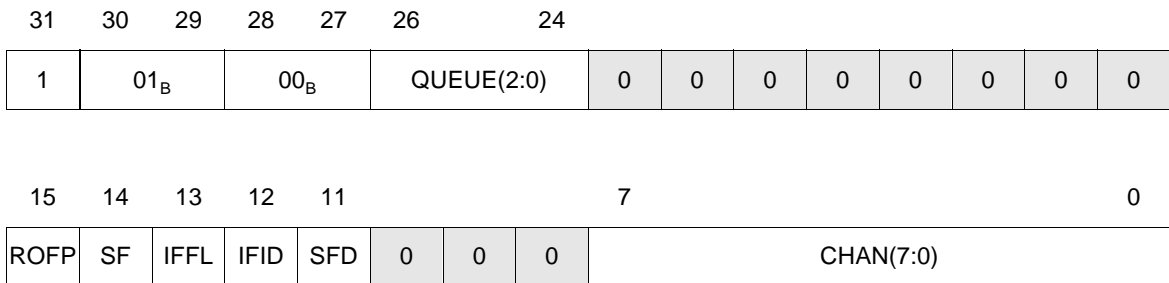
Channel interrupt are divided into two subtypes:

- Receive Interrupt I and Transmit Interrupt I
- Receive Interrupt II and Transmit Interrupt II

Subtype I contains interrupts which indicate the general status of a channel. These interrupts are not linked to a descriptor.

Subtype II contains interrupts which indicate a channel or packet status that is linked to a descriptor. Each interrupt vector contains a descriptor ID which can be used for tracking purposes.

Receive Interrupt I



- ROFP** Receive Buffer Overflow
The 'Receive Buffer Overflow' interrupt vector is generated, when one or more whole frames or short frames or changes of interframe time-fill (HLDC, PPP) or data in general (TMA) has been discarded due to the inaccessibility of the internal receive buffer.
- SF** Short Frame Detected
The 'Short Frame Detected' interrupt vector is generated, when the receiver detected a frame which length matches the condition defined in CONF1.SFL.
- IFFL** Interframe Time-fill Flag
The 'Interframe Time-fill Flag' interrupt vector is generated, when the receiver detected a interframe time-fill change from FF_H to 7E_H.
- IFID** Interframe Time-fill Idle
The 'Interframe Time-fill Idle' interrupt vector is generated, when the receiver detected a interframe time-fill change from 7E_H to FF_H.

Functional Description

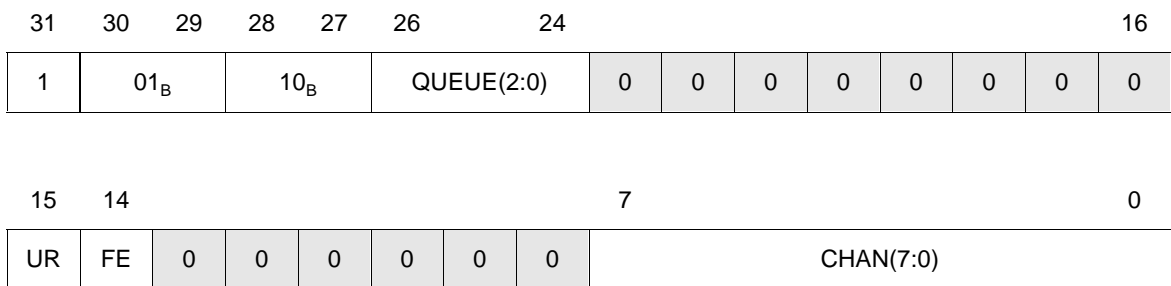
SFD Small Frames Dropped

The 'Small Frames Dropped' interrupt vector is generated, when the receiver discarded N small frames. The length of small frames is defined in CONF3.MINFL and the threshold value N is defined in register SFDT.

CHAN Channel Number

This bit field identifies the channel for which the information in the interrupt vector is valid.

Transmit Interrupt I



UR Underrun

The 'Underrun' interrupt vector is generated, when the transmit buffer was not able to provide data to the protocol machine transmit. If this happens during transmission of a HDLC or PPP packet, the transmitter will end the already started data packet with an abort sequence.

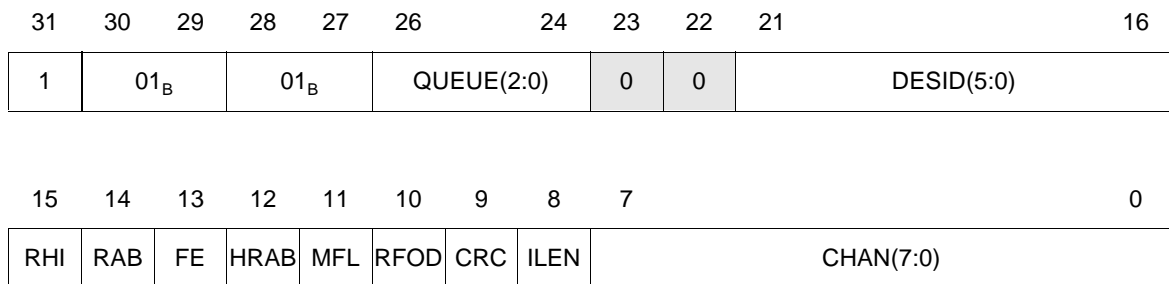
FE Frame End

The 'Frame End' interrupt vector is generated, when one complete data packet has been transmitted via serial side.

CHAN Channel Number

This bit field identifies the channel for which the information in the interrupt vector is valid.

Receive Interrupt II

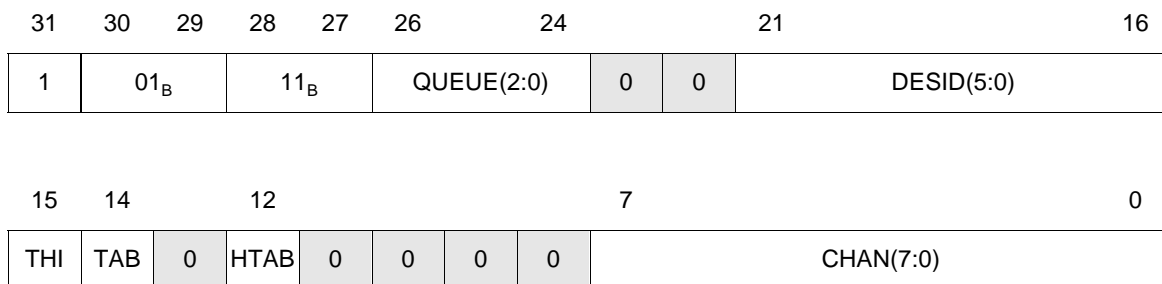


- CHAN** Channel Number
This bit field identifies the channel for which the information in the interrupt vector is valid.
- RHI** (Receive) Host Initiated Interrupt
The '(Receive) Host Initiated' interrupt vector will be issued, if bit RHI is set in a receive descriptor and processing of this descriptor has finished. After receiving this interrupt vector, system software can release the descriptor, e.g. put the descriptor into a free pool.
- RAB** Receive Abort
The 'Receive Abort' interrupt vector is generated, when an incoming data packet is aborted (more than 6 '1' in case of HDLC or more than 15 '1' in case of PPP) or if the receiver got a receive abort command from the system CPU.
- FE** Frame End
The 'Frame End' interrupt Vector is generated, when one complete frame has been received completely and has been stored in system memory.
- HRAB** Hold Caused Receive Abort
The 'Hold Caused Receive Abort' interrupt vector is generated, when the receiver discarded the first data packet after it has found a HOLD bit in a receive descriptor.
- RAB, HRAB** Silent Discard
The 'Silent Discard' interrupt vector (bit RAB and HRAB set together) occurs, if two or more frames have been discarded by the receiver due to continuous inaccessibility of receive descriptor. This occurs, if receive descriptor has HOLD bit set and receiver gets further data packets. The interrupt vector will be generated for each packet discarded.

Functional Description

- MFL** Maximum Frame Length Exceeded
The 'Maximum Frame Length Exceeded' interrupt vector is generated, when the length of a received data packet exceeded the frame length defined in CONF1.MFL.
- RFOD** Receive Frame Overflow DMA
The 'Receive Frame Overflow DMA' interrupt indicates that protocol handler was unable to transfer data to the receive buffer. As soon as receive buffer can store data again, this interrupt is generated.
- CRC** CRC Error
The 'CRC Error' interrupt vector is generated, when the internally calculated CRC and the CRC of a received packet did not match.
- ILEN** Invalid Length
The 'Invalid Length' interrupt vector is generated, when the bit length of received frame was not divisible by 8.

Transmit Interrupt II



- DESID** Descriptor ID
This bit field is a copy of the descriptor ID of the transmit descriptor which is currently in use. It can be used for tracking purposes.
- THI** (Transmit) Host Initiated Interrupt
The '(Transmit) Host Initiated' interrupt vector is generated, if bit THI is set in a transmit descriptor and processing of this descriptor has finished. After receiving this interrupt vector, system software can release the descriptor, e.g. put the descriptor into a free pool.
- TAB** Transmit Abort
The 'Transmit Abort' interrupt vector is generated, either when the 'Transmit Abort/Branch' command was given and therefore one frame could not be transmitted completely or when NO and FE were set to 0 in a transmit descriptor and previous frame was incompletely specified.

Functional Description

| | |
|------|---|
| HTAB | <p>Hold Caused Transmit Abort</p> <p>The 'Hold Caused Transmit Abort' interrupt vector is generated, when data management unit retrieved a transmit descriptor where HOLD was set and FE equals 0. The interrupt will be generated after the data section was transferred completely. After transmission of frame based protocols (HDLC, PPP) protocol machine appends abort sequence due to incomplete packet.</p> |
| CHAN | <p>Channel Number</p> <p>This bit field identifies the channel for which the information in the interrupt vector is valid.</p> |

4.13.1.5 Command Interrupts

Command interrupts are written to the command interrupt queue (interrupt queue eight).

Transmit Interrupts

| | | | | | | | | | | | | | | | | | |
|----|-------------------|---|---|----|---|---|---|---|-----------|---|---|---|----|----|---|-----|-----|
| 31 | 30 | | | 27 | | | | | | | | | 17 | 16 | | | |
| 1 | 0010 _B | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TCF | TCC |
| | | | | | | | | 7 | | | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CHAN(7:0) | | | | | | | | |

- TCF** Transmit Command Failed
- The 'Transmit Command Failed' interrupt vector is issued, if the command 'Transmit Init' given via register *CSPEC_CMD.XCMD* could not be finished. This happens, when
- system software tried to allocate more buffer locations for a channel than were available.
 - system software specified thresholds (transmit forward threshold, transmit refill threshold), which were greater than the specified transmit buffer size.
- Note: The sum of both thresholds must be smaller than the transmit buffer size of a particular channel. Erroneous programming does NOT result in the 'Transmit Command Failed' interrupt vector.*
- TCC** Transmit Command Complete
- The 'Transmit Command Complete' interrupt vector is issued after successful completion of commands 'Transmit Init' and 'Transmit Off', which can be issued via register *CSPEC_CMD.XCMD*.
- CHAN** Channel Number
- This bit field contains the channel number of the affected channel.

Receive Interrupts

| | | | | | | | | | | | | | | |
|----|-------------------|----|---|---|---|---|---|---|---|---|---|---|----|-----|
| 31 | 30 | 27 | | | | | | | | | | | 16 | |
| 1 | 0000 _B | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RCC |

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|-----------|--|--|--|--|--|--|--|---|
| 15 | | | | | | | | 7 | | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | CHAN(7:0) | | | | | | | | |

- RCC** Receive Command Complete
 The 'Receive Command Complete' interrupt vector is issued after successful completion of commands 'Receive Init' and 'Receive Off', which can be issued via register *CSPEC_CMD.RCMD*.
- CHAN** Channel Number
 This bit field contains the channel number of the affected channel.

4.13.2 Layer One Interrupts

All layer one related interrupts, that is interrupts issued by either the T1/E1 framer, the M13 multiplexer and DS2/DS3 framer, the facility data link or the PCI to Local Bus mailbox, are stored in an internal interrupt FIFO which is located inside the MUNICH256FM and can be read from either the local microprocessor or (for test purposes) via the chip internal bridge from the host processor located on the PCI bus.

The T1/E1 framer, the facility data link, the M13 multiplexer and DS2/DS3 framer, and the mailbox forward their specific interrupts to the internal interrupt FIFO. The interrupt FIFO triggers the $\overline{\text{LINT}}$ pin which indicates that there is at least one interrupt vector available. The interrupt FIFO then can be read from either PCI side or local bus side. The interrupt vector contains a coding for the interrupt reason and a last indication when there is no further interrupt vector stored in the internal interrupt FIFO. The interrupts of the internal layer one interrupt FIFO or the local bus interrupt $\overline{\text{LINT}}$ can also be reported via pin INTA.

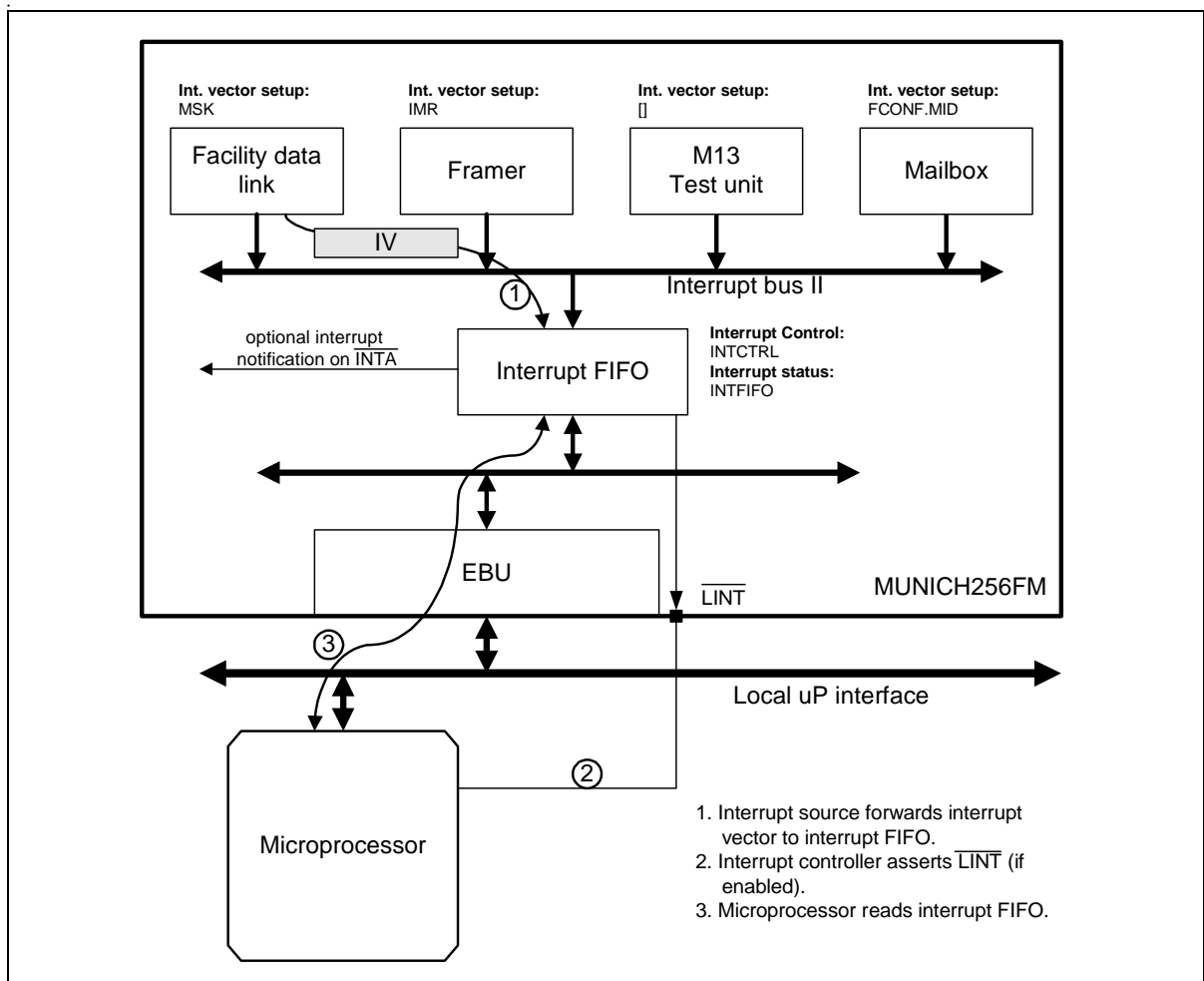
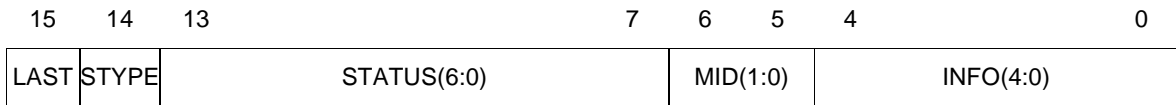


Figure 4-19 Framer, M13 and Facility Data Link and Mailbox Interrupt Notification

4.13.2.1 General Interrupt Vector Structure



- LAST** Last indication
- LAST indicates that at least one more valid interrupt vector is stored in the internal interrupt FIFO. This bit is generated at read access time.
- 0 There is at least one more interrupt in the internal interrupt FIFO.
- 1 This interrupt is the last interrupt that is stored in the internal interrupt FIFO.
- STYPE** Subtype of interrupt vector
- This bit is used to indicate different subtypes of interrupt vectors.
- STATUS** Interrupt status
- The interrupt status depends on STYPE and MID. Please refer to the detailed description of the interrupt vectors in the next chapters.
- MID** Module ID
- The bit field identifies the interrupt source.
- 00_B T1/E1 Framer Interrupts
- 01_B Facility Data Link Interrupts
- 10_B M13 Multiplexer and DS2/DS3 framer Interrupts
- 11_B Mailbox Interrupt
- INFO** Information
- The content of this bit field contains further information about the interrupt, e.g. the affected port.

4.13.2.2 T1/E1 Framer Interrupts

The framer interrupts are divided into type 0 and type I interrupts. The distinction is made in bit 14 of the interrupt vector.

Interrupt Type 0

| | | | | | | | | | | | | |
|------|----|------|------|-----|----|-----|------|-------|-----------------|-----------|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 0 |
| LAST | 0 | AISS | LOSS | RAS | ES | SEC | LLBS | PRBSS | 00 _B | PORT(4:0) | | |

Interrupt Type I

| | | | | | | | | | | | | |
|------|----|---|---|------|-----|----------|-----|------|-----------------|-----------|---|---|
| 15 | 14 | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 0 |
| LAST | 1 | 0 | 0 | T400 | CRC | PDEN/AUX | FAS | MFAS | 00 _B | PORT(4:0) | | |

- AISS** Alarm Indication Signal Status
The 'Alarm Indication Signal Status' interrupt vector is generated, whenever the MUNICH256FM detects a change in the alarm indication. The actual state, i.e. active/not active, is shown in FRS.AIS.
- LOSS** Loss of Signal Status
The 'Loss of Signal Status' interrupt vector is generated, whenever the MUNICH256FM detects a change in FRS.LOS.
- RAS** Remote Alarm Status
The 'Remote Alarm Status' interrupt vector is generated, whenever the MUNICH256FM received remote alarm status changes. The actual state, i.e. active/not active, is shown in FRS.RRA.
- ES** Errored Second
The 'Errored Second' interrupt vector is generated for the first errored second event in a time interval of one second. Errored second events are:
1. Loss of frame alignment (this includes indirectly AIS or Loss of Signal)
2. CRC error received (CRC-6 or CRC-4).
- SEC** One Second Tick
The 'One Second Tick' interrupt vector is generated, when the internal one second timer has expired. The timer is derived from the incoming receive clock of the corresponding port.

Functional Description

| | |
|----------|--|
| LLBS | <p>Line Loopback Status</p> <p>The 'Line Loopback Status' interrupt vector is generated, whenever the MUNICH256FM detects a change in either the line loopback deactuation signal or the line loopback actuate signal. The actual state of the signals is shown in FRS.LLBDD and FRS.LLBAD.</p> |
| PRBS | <p>PRBS Status</p> <p>The 'PRBS Status' interrupt vector is generated, whenever the MUNICH256FM synchronization state of the PRBS receiver changes. The actual state of the receiver, i.e. synchronized/not synchronized, is shown in FRS.PRBS.</p> |
| T400 | <p>400 Millisecond</p> <p>This interrupt vector is generated when the framer has found the double framing (basic framing) and is searching for the multiframing. This interrupt vector will be generated to indicate that no multiframing could be found within a time window of 400 ms after basic framing has been achieved. In multiframe synchronous state this interrupt will not be generated.</p> |
| CRC | <p>Receive CRC Error</p> <p>This interrupt vector is generated, when the CRC-6 checksum of an T1 ESF multiframe or the CRC-4 checksum of an E1 CRC-4 multiframe was incorrect.</p> |
| PDEN/AUX | <p>Pulse Density Violation Detected / Auxiliary Pattern Detected</p> <p>This interrupt vector is generated, whenever the MUNICH256FM detects a change in bit FRS.PDEN/AUX. Bit PDEN/AUX is set whenever bit FRS.PDEN.AUX toggles.</p> |
| FAS | <p>Frame Alignment Status</p> <p>The 'Frame Alignment Status' interrupt vector is generated, whenever the MUNICH256FM detects a change in frame alignment. The actual state, i.e. aligne/not aligned, is shown in bit FRS.LFA.</p> |
| MFAS | <p>Multiframe Alignment Status</p> <p>The 'Multiframe Alignment Status' interrupt vector is generated, whenever the MUNICH256FM detects a change in multiframe alignment. The actual state, i.e. aligned/not aligned, is shown in bit FRS.LMFA.</p> |
| PORT | <p>Port Number</p> <p>0..27 The port number the interrupt vector is associated with.</p> |

4.13.2.3 Facility Data Link Interrupts

Receive Interrupts

| | | | | | | | | | | | | | | |
|------|----|---|---|-----|-----|-----|-----|-----|-----------------|-----------|---|--|--|---|
| 15 | 14 | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | | | 0 |
| LAST | 0 | 0 | 0 | RSA | SSM | RPF | RME | ISF | 01 _B | PORT(4:0) | | | | |

- RSA** Receive S_a Data Valid
S_a data in RSAW1 - RSAW3 is valid.
- SSM** SSM Data Valid
This bit is set, when a new synchronization status message has been received. The synchronization status message is stored in register RSAW4.
- RPF** Receive Pool Full
This bit is set, when 32 bytes of a frame have been received and are stored in the receive FIFO. The frame is not yet completely received.
- RME** Receive Message End
This bit is set, when one complete message of length less than 32 bytes or the last part of a frame at least 32 bytes long is stored in the receive FIFO. The number of bytes in RFF.RFIFO can be determined reading the port status register PSR.
- ISF** Incorrect Synchronization Format
This bit is set, when no eight consecutive '1's are detected within 32 bits in BOM mode. Only valid if BOM receiver has been activated.
- PORT** Port Number
0..27 The port number the interrupt vector is associated with.

Transmit Interrupts

| | | | | | | | | | | | | | | |
|------|----|---|---|---|------|------|-----|-----|-----------------|---|-----------|--|--|---|
| 15 | 14 | | | | 10 | 9 | 8 | 7 | 6 | 5 | 4 | | | 0 |
| LAST | 1 | 0 | 0 | 0 | TXSA | ALLS | XDU | XPR | 01 _B | | PORT(4:0) | | | |

| | |
|------|--|
| TXSA | Transmit S _a Data Sent |
| | The 'Transmit S _a Data Sent' is generated, when S _a data stored in XSAW1 - XSAW3 has been sent N times, where N is defined prior to transmission in XSAW3.XSAV. |
| ALLS | All Sent |
| | The 'All Sent' interrupt vector is generated, when the last bit of a frame to be transmitted is completely sent out and XFF.XFIFO is empty. |
| XDU | Transmit Data Underrun |
| | The 'Transmit Data Underrun' interrupt vector is generated, when the transmit FIFO runs out of data during transmission of a frame. The signalling controller terminates the affected frame with an abort sequence. |
| XPR | Transmit Pool Ready |
| | The 'Transmit Pool Ready' interrupt vector is generated, when a new data block of up to 32 bytes can be written to transmit FIFO. 'Transmit Pool Ready' is the fastest way to access the transmit FIFO. It has to be used for transmission of long frames, back-to-back frames or frames with shared flag. |
| PORT | Port Number |
| | 0..27 The port number the interrupt vector is associated with. |

4.13.2.4 DS3, DS2 and Test Unit Interrupts

Note: The DS3, DS2 and test unit interrupts are separated by the INFO field (bits 4 through 0).

DS3 Interrupts Type 0

| | | | | | | | | | | | | |
|------|----|-----|------|-------|------|------|------|-----|-----------------|--------------------|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 0 |
| LAST | 0 | AIC | XBIT | IDLES | AISS | REDS | LOSS | FAS | 10 _B | 00111 _H | | |

DS3 Interrupts Type 1

| | | | | | | | | | | | | |
|------|----|----|------|------|------|------|-----|----------------|-----------------|--------------------|---|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 0 |
| LAST | 1 | 0 | CLKS | RSDL | TSDL | LPCS | SEC | N _r | 10 _B | 00111 _H | | |

- CLKS** DS3 Clock Status
- The 'DS3 Clock Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the transmit clock or the receive clock, i.e. clock is activated/deactivated. The actual status of the clock is shown in D3RSTAT.LRXC and D3RSTAT.LTXC.
- RSDL** Receive Spare Data Link Transfer Buffer Full
- The 'Receive Spare Data Link Transfer Buffer Full' interrupt vector is generated when the receive spare data link buffer needs to be emptied.
- TSDL** Transmit Spare Data Link Transfer Buffer Empty
- The 'Transmit Spare Data Link Transfer Buffer Empty' interrupt vector is generated when the transmit spare data link buffer needs to be filled.
- LPCS** Loopback Code Status
- The 'Loopback Code Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the received loopback codes. Actual loopback codes can be found in register D3RLPCS.
- SEC** 1 Second Interrupt
- The '1 Second Interrupt' is generated every second.
- N_r** Received new N_r-Bit
- The 'Received new N_r-Bit' interrupt vector is generated whenever the MUNICH256FM detects a change in the NA overhead bits and when its state is persistent for at least three multiframes.

Functional Description

| | |
|-------|---|
| AIC | Received new AIC-Bit The 'Received new AIC-Bit' interrupt vector is generated whenever the MUNICH256FM detects a change in the AIC overhead bits and when its state is persistent for at least three multiframes. |
| XBIT | Received X-Bit The 'Received new X-Bit' interrupt vector is generated whenever the MUNICH256FM detects a change in the X overhead bits and when its state is persistent for at least three multiframes. |
| IDLES | DS3 Idle Signal Status The 'DS3 Idle Signal Status' interrupt vector is generated whenever the MUNICH256FM detects a change of the idle signal. D3RSTAT.IDLES contains the actual state of the idle state, i.e. active/not active. |
| AISS | DS3 Alarm Indication Signal Status The 'DS3 Alarm Indication Signal Status' is generated whenever the MUNICH256FM detects a change in the AIS alarm state. D3RSTAT.AISS shows the actual AIS alarm state, i.e. active/not active. |
| REDS | DS3 Red Alarm Status The 'DS3 Red Alarm' interrupt vector is generated whenever the MUNICH256FM detects a change in the red alarm state. D3RSTAT.RED shows the actual red alarm state, i.e. active/not active. |
| LOSS | DS3 Input Signal Status The 'DS3 Input Signal Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the DS3 input signal state, i.e. loss/no loss. D3RSTAT.LOSS shows the actual state of the DS3 input signal. |
| FAS | DS3 Frame Alignment Status The 'DS3 Frame Alignment Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the DS3 frame alignment. D3RSTAT.FAS shows the actual state. |

DS2 Framer Interrupts

Note: The effected DS2 tributary is encoded in the INFO field (bits 4..0).

| | | | | | | | | | | | | | |
|------|----|---|------|------|------|-----|-----|-----|-----------------|---|---|--|---|
| 15 | 14 | | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | | 0 |
| LAST | 0 | 0 | LPCS | AISS | REDS | RES | RAS | FAS | 10 _B | 00000 _H - 00110 _H | | | |

- LPCS** Loop Code Status
The 'Loopback Code Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the received loopback codes. Actual loopback codes can be found in register D2RLPCD.
- AISS** DS2 Alarm Indication Signal Status
The 'DS2 Alarm Indication Signal Status' is generated whenever the MUNICH256FM detects a change in the AIS alarm state. D2RSTAT.AIS shows the actual AIS alarm state, i.e. active/not active.
- REDS** DS2 Red Alarm Status
The 'DS2 Red Alarm Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the red alarm state. D3RSTAT.RED shows the actual red alarm state, i.e. active/not active.
- RES** Received new Reserved ITU-T G.747 Overhead Bit
The 'Received new Reserved ITU-T G.747 Overhead Bit' interrupt vector is generated whenever the MUNICH256FM detects a change in the reserved ITU-T G.747 overhead bit and when its state is persistent for at least three multiframes. D2R[.] shows the actual state of the overhead bit.
- RAS** Remote Alarm Status
The 'Remote Alarm Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the remote alarm indication and when its state is persistent for at least three multiframes. D2RSTAT.RA shows the actual state of the remote alarm indication.
- FAS** DS2 Frame Alignment Status
The 'DS2 Frame Alignment Status' interrupt vector is generated whenever the MUNICH256FM detects a change in the DS2 frame alignment. D2RSTAT.LFA shows the actual status of frame alignment.

Test Unit Interrupts Type 0

| | | | | | | | | | | | | | | |
|------|----|---|---|-----|-----|----|----|-----|-----------------|--------------------|---|--|--|---|
| 15 | 14 | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | | | 0 |
| LAST | 0 | 0 | 0 | EMI | LBE | A1 | A0 | OOS | 10 _B | 01000 _H | | | | |

- OOS** Receiver Out Of Synchronization
The 'Receiver Out of Synchronization' interrupt vector is generated whenever the test unit detects a change in synchronization. The actual state of the receiver is shown in TURSTAT.OOS.
- A0** Input all '0's
The 'Input all '0's' interrupt vector is generated whenever the MUNICH256FM detects 32 continuous '0's or when this condition is resolved. The actual state is shown in TURSTAT.A0.
- A1** Input all '1's
The 'Input all '1's' interrupt vector is generated whenever the MUNICH256FM detects 32 continuous '1's or when this condition is resolved. The actual state is shown in TURSTAT.A1.
- LBE** Latched Bit Error Detected Flag
The 'Latched Bit Error Detected Flag' interrupt vector is generated with the first occurrence of a bit error.
- EMI** End of Measurement Interval
The 'End of Measurement Interval' interrupt vector is generated when the end of the programmed measurement interval is reached.

4.13.2.5 Mailbox Interrupts

| | | | | | | | | | | | | |
|------|----|-------------|--|--|--|---|-----------------|--------------------|---|--|--|---|
| 15 | 14 | 13 | | | | 7 | 6 | 5 | 4 | | | 0 |
| LAST | 0 | STATUS(6:0) | | | | | 11 _B | 00000 _B | | | | |

The 'Mailbox' interrupt vector is generated, in case that the host CPU on PCI side has written data to the mailbox status register MBP2E0. The bit field STATUS contains a copy of MBE2P0.MB(6:0).

5 Interface Description

5.1 PCI Interface

A 32-bit and 66 MHz capable PCI bus controller provides the interface between the MUNICH256FM and the host system. PCI Interface pins are measured as compliant to the 3.3V signalling environment according to the PCI specification Rev. 2.1.

The PCI bus controller operates as initiator or target. Commands are supported as follows:

- Master memory read single DWORD/burst of up to 64 DWORDs with zero wait cycles.
- Master memory write single DWORD/burst of up to 64 DWORDs with zero wait cycles.
- Slave memory read single DWORD.
- Slave memory write single DWORD.

Fast back-to-back transfers are provided for slave accesses only. All read/write accesses to the MUNICH256FM must be 32-bit wide, that is all bytes must be enabled. Non 32-bit accesses result in system interrupt.

Refer also to the PCI specification Rev. 2.1 for detailed information about PCI bus protocol.

5.1.1 PCI Read Transaction

The transaction starts with an address phase which occurs during the first cycle when $\overline{\text{FRAME}}$ is activated (clock 1 in **Figure 5-1**). During this phase the bus master (initiator) outputs a valid address on AD(31:0) and a valid bus command on C/BE (3:0). The first clock of the first data phase is clock 3. During the data phase C/ $\overline{\text{BE}}$ indicate which byte lanes on AD(31: 0) are involved in the current data phase.

The first data phase on a read transaction requires a turnaround cycle. In **Figure 5-1** the address is valid on clock 2 and then the master stops driving AD. The target drives the AD lines following the turnaround when $\overline{\text{DEVSEL}}$ is asserted. ($\overline{\text{TRDY}}$ cannot be driven until $\overline{\text{DEVSEL}}$ is asserted.) The earliest the target can provide valid data is clock 4. Once enabled, the AD output buffers of the target stay enabled through the end of the transaction.

A data phase may consist of a data transfer and wait cycles. A data phase completes when data is transferred, which occurs when both $\overline{\text{IRDY}}$ and $\overline{\text{TRDY}}$ are asserted. When either is deasserted a wait cycle is inserted. In the example below, data is successfully transferred on clocks 4, 6 and 8, and wait cycles are inserted on clocks 3, 5 and 7. The first data phase completes in the minimum time for a read transaction. The second data phase is extended on clock 5 because $\overline{\text{TRDY}}$ is deasserted. The last data phase is extended because $\overline{\text{IRDY}}$ is deasserted on clock 7. The Master knows at clock 7 that the next data phase is the last. However, the master is not ready to complete the last

Interface Description

transfer, so $\overline{\text{IRDY}}$ is deasserted on clock 7, and $\overline{\text{FRAME}}$ stays asserted. Only when $\overline{\text{IRDY}}$ is asserted can $\overline{\text{FRAME}}$ be deasserted, which occurs on clock 8.

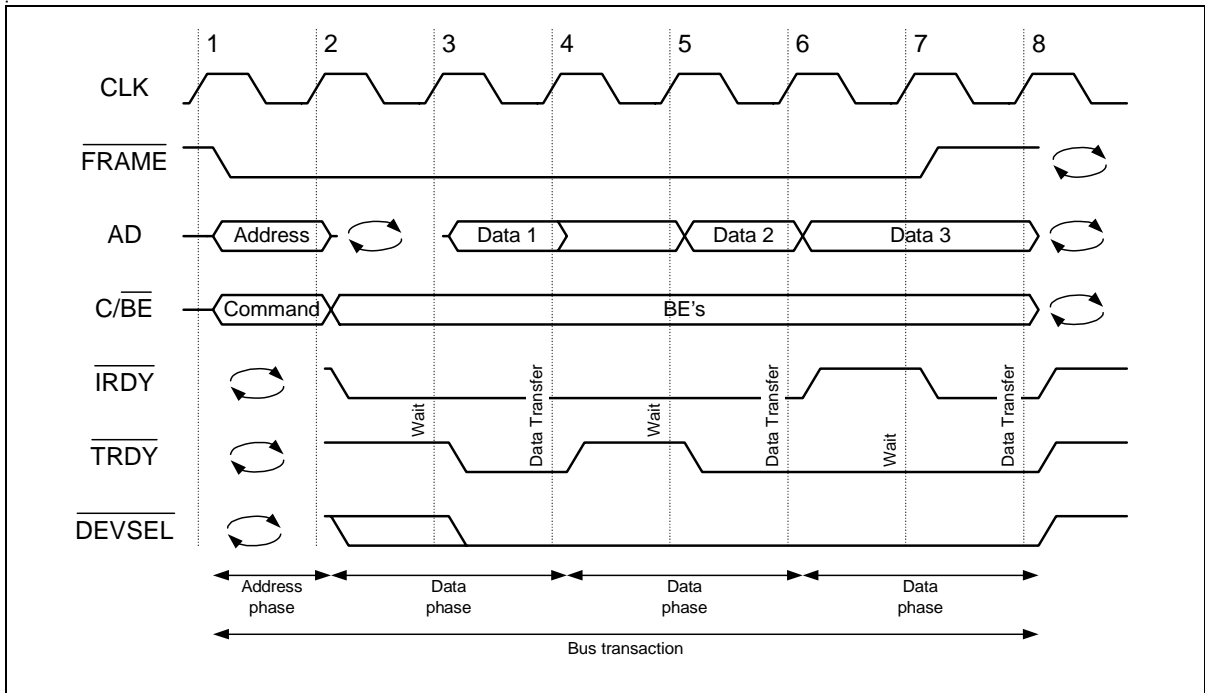


Figure 5-1 PCI Read Transaction

5.1.2 PCI Write Transaction

The transaction starts when $\overline{\text{FRAME}}$ is activated (clock 1 in **Figure 5-2**). A write transaction is similar to a read transaction except no turnaround cycle is required following the address phase. In the example, the first and second data phases complete with zero wait cycles. The third data phase has three wait cycles inserted by the target. Both initiator and target insert a wait cycle on clock 5. In the case where the initiator inserts a wait cycle (clock 5), the data is held on the bus, but the byte enables are withdrawn. The last data phase is characterized by $\overline{\text{IRDY}}$ being asserted while the $\overline{\text{FRAME}}$ signal is deasserted. This data phase is completed when $\overline{\text{TRDY}}$ goes active (clock 8).

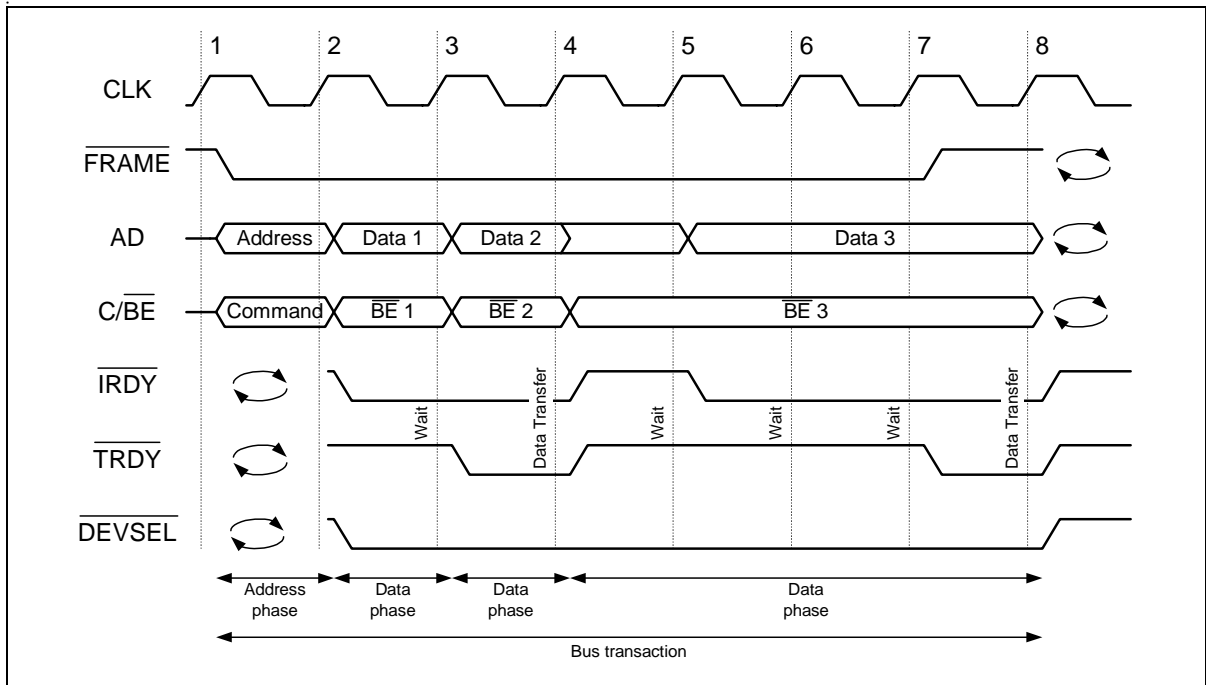


Figure 5-2 PCI Write Transaction

5.2 SPI Interface (ROM Load Unit)

Additional pins, which are not covered from the PCI specification, but are closely related, are the SPI pins. Via the SPI pins the vendor ID and the vendor subsystem ID can be loaded into the corresponding PCI configuration registers during start-up of the device.

The SPI Interface supports EEPROMs with an eight bit address space.

After a system reset, the MUNICH256FM starts reading the first byte out of the connected EEPROM at address 00_H . If this byte is equal AA_H , the device continues reading out the memory contents. Everytime four bytes are read out of the EEPROM (starting with byte address 01_H), the EEPROM interface writes the read information to the PCI configuration space. The first four bytes will be written to the PCI configuration space address 00_H , the next four bytes to the PCI configuration space address 04_H and so on. So the contents of the EEPROM, starting with EEPROM byte address 01_H , will be mapped over the PCI configuration space after a system reset. During this configuration phase, all accesses to the PCI interface will be answered with 'retry' by the PCI interface.

If the first byte in the EEPROM is not equal AA_H , the EEPROM interface stops loading the PCI configuration space immediately, and the PCI interface can be accessed. The PCI configuration space in this case contains the default values.

The configuration mechanism through the serial interface can be disabled by pin SPLOAD. If this pin is connected to '0', the configuration mechanism is disabled. The

bridge can be accessed through the PCI Interface directly after a system reset. In this case the PCI configuration space contains the default values.

5.2.1 Accesses to a SPI EEPROM

The EEPROM contents can also be controlled (read and write) by the software. For this, a special EEPROM control register is implemented as part of the PCI configuration space. To start a read/write transaction to an connected EEPROM, you have to set the command, the byte address (for read-/write data commands), the data to be written and the start indication by writing to the EEPROM control register SPI in the PCI configuration space. If the interface detects SPI.START asserted (= '1'), it interprets the command and starts the read-/write transaction to the connected EEPROM. After the transaction has finished, the EEPROM control module deasserts the start bit. If the command was a read command (Read Status Register, Read Data from Memory Array), the byte that was read out of the EEPROM is available in the data register. For transactions started with the EEPROM Control register, the interface does not check if an EEPROM is connected to the SPI bus, because the EEPROM is full passive. A full functional description of the SPI commands and their usage as well as a description of the EEPROMs status register can be found in the description of the EEPROM that will be selected by a board vendor.

Byte Address

For read and write transaction to the connected EEPROM, the byte address must be written in this register before the transaction is started.

Data

For the write status register transaction and the write data to memory array transactions, the data that has to be written to the EEPROM must be written to this register before the transaction is started. After a read status register transaction or a read data from memory array transaction has finished (Bit SPI.START is deasserted), the byte received from the EEPROM is available in this register.

Start

To start the EEPROM transaction defined via register SPI the bit SPI.START must be set to '1' by a write transaction through the PCI interface. After the transaction is finished, the EEPROM start bit is deasserted by the EEPROM interface controller. This signal has to be polled by system software.

5.2.2 SPI Read Sequence

The MUNICH256FM selects an external EEPROM by pulling $\overline{\text{SPCS}}$ low. The eight bit read sequence is transmitted followed by the eight bit address. After the read instruction

and address is sent, the data stored in the memory at the selected address is shifted in on the SPSI pin. The read operation is terminated by setting $\overline{\text{SPCS}}$ high (see **Figure 5-3**).

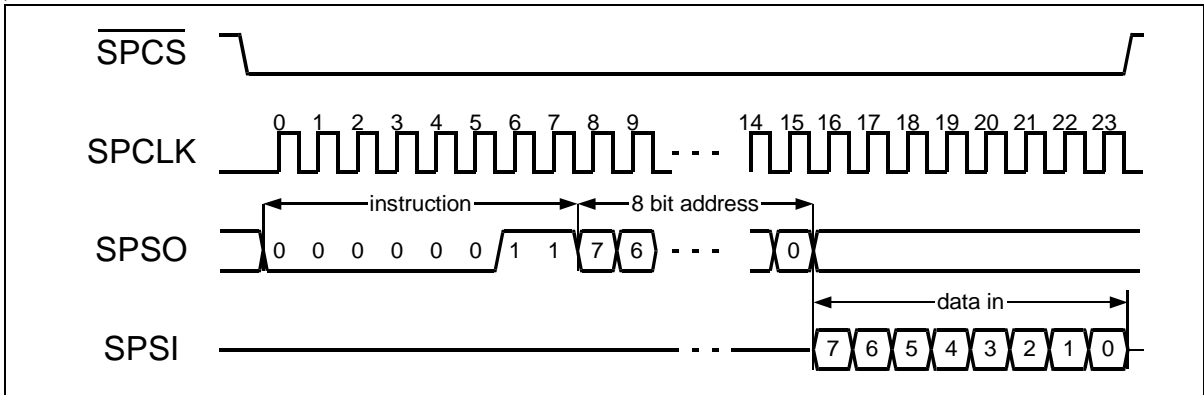


Figure 5-3 SPI Read Sequence

5.2.3 SPI Write Sequence

Prior to any attempt to write data to an external EEPROM, the write enable latch must be set by issuing the WREN instruction. This is done by setting $\overline{\text{SPCS}}$ low and then clocking out the WREN instruction. After all eight bits of the instruction are transmitted, the SPCS will be brought high to set the write enable latch.

Once the write enable latch is set, the user may proceed by issuing a write instruction, followed by the eight bit address and then the data to be written. In order that data will actually be written to the EEPROM, the SPCS is set high after the least significant bit (D0) of the data byte has been clocked in. Refer to **Figure 5-4** for detailed illustrations on the byte write sequence. While the write is in progress, the register bit SPI.START may be read to check the status of the transaction. When a write cycle is completed, the register bit SPI.START is reset.

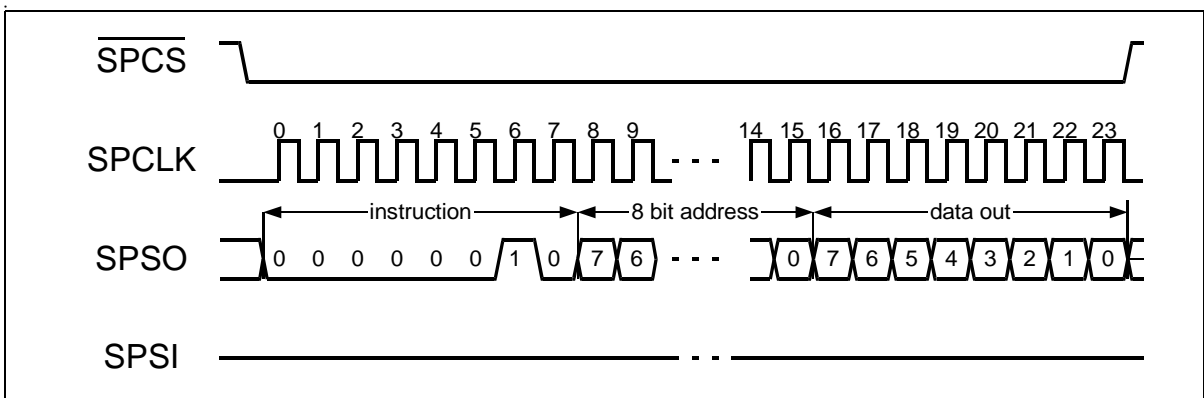


Figure 5-4 SPI Write Sequence

5.3 Local Microprocessor Interface

The Local Microprocessor Interface is a demultiplexed switchable Intel or Motorola style interface with master and slave functionality. In slave mode it is used to operate the M13 multiplexer, DS3/DS2 framer, T1/E1 framer and the facility data link of the MUNICH256FM. The MUNICH256FM provides a local clock output LCLK, which is a feed through of the PCI system clock as clock reference for the local microprocessor interface. The local bus master capability allows to access peripherals located on the local bus via the PCI interface. Bit FCONF.LME enables the bus master capability.

The base address register two is disabled per default and can be enabled during start-up of the internal PCI interface. This is done by setting bit MEM.BAR2 in the PCI configuration space.

The MUNICH256FM supports a maximum of three 8 kByte pages of memory on the local address bus. The correspondence between the accessed PCI memory space (mapped via base address register 2) and the asserted chip selects is shown in table 5-1. The mapping of the PCI byte enables to the local bus address is dependent on the selected bus mode and is explained in detail in the corresponding section.

Table 5-1 Correspondence between PCI memory space and chip select

| Page | AD(14:0) | $\overline{\text{LCS2}}$ | $\overline{\text{LCS1}}$ |
|------|---------------------------------------|--------------------------|--------------------------|
| 0 | 0000 _H - 1FFF _H | 1 | 0 |
| 1 | 2000 _H - 3FFF _H | 0 | 1 |
| 2 | 4000 _H - 5FFF _H | 0 | 0 |
| 3 | 6000 _H - 7FFF _H | Not valid | |

5.3.1 Intel Mode

5.3.1.1 Slave Mode

In Intel slave mode the bus interface supports 16-bit transactions in demultiplexed bus operation. It uses the local bus port pins LA(12:1) for the 16 bit address and the local bus port pins LD(15:0) for 16 bit data. A read/write access is initiated by placing an address on the address bus and asserting LCS0 (**Figure 5-5**). The external processor then activates the respective command signal (LRD, LWR). Data is driven onto the data bus either by the MUNICH256FM (for read cycles) or by the external processor (for write cycles). After a period of time, which is determined by the access time to the internal registers valid data is placed on the bus, which is indicated by asserting the active low signal LRDY.

Note: LCS0 need not be deasserted between two subsequent cycles to the same device.

Read cycles

Input data can be latched and the command signal can be deactivated now. This causes the MUNICH256FM to remove its data from the data bus which is then tri-stated again. LRDY is driven high and will be tri-stated as soon as LCS0 is deasserted.

Write cycles

The command signal can be deactivated now. If a subsequent bus cycle is required, the external processor can place the respective address on the address bus.

5.3.1.2 Master Mode

A read/write access from the PCI bus to the 16 bit demultiplexed local bus is initiated by accessing the PCI memory space base which is controlled by the base address register 2. Each valid read or write access to this base address triggers the local bus master interface which in turn starts arbitration for the local bus by asserting LHOLD (see (1) in **Figure 5-6**). As soon as the MUNICH256FM gets access to the local bus (LHLDA asserted) it starts the local bus latency timer and begins a read/write transaction as the bus master. The signal LHOLD remains asserted while a transaction is in progress or as long as the local bus latency timer is not expired. A read/write transaction begins when the MUNICH256FM places a valid address on the address bus, sets the LBHE signal which indicates a 8- or 16-bit bus access and asserts the chip select signals LCS1 and/ or LCS2. Then the MUNICH256FM activates the respective command signals (LRD, LWR). Data is driven onto the data bus either by the MUNICH256FM (for write cycles) or by the accessed device (for read cycles).

A transaction is finished on the local bus when the external device asserts LRDY (ready controlled bus cycles) or when the internal wait state timer expires.

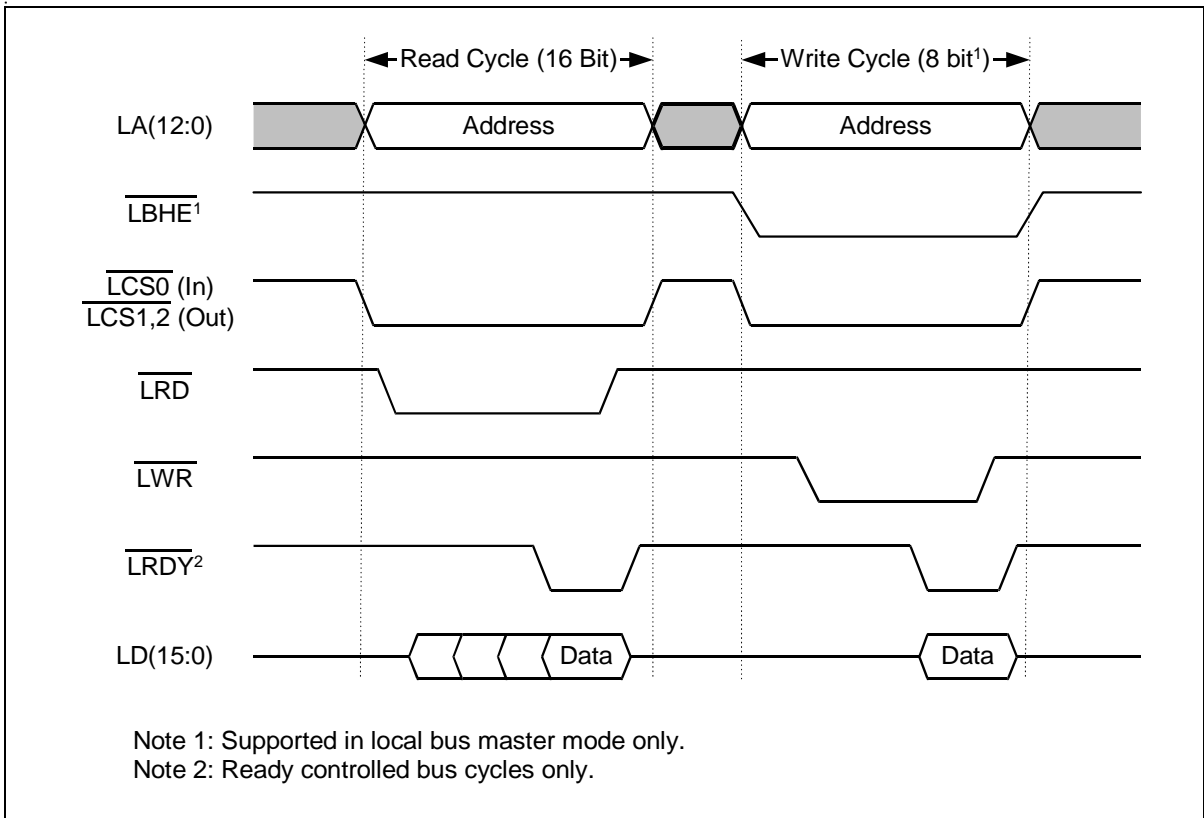


Figure 5-5 Intel Bus Mode

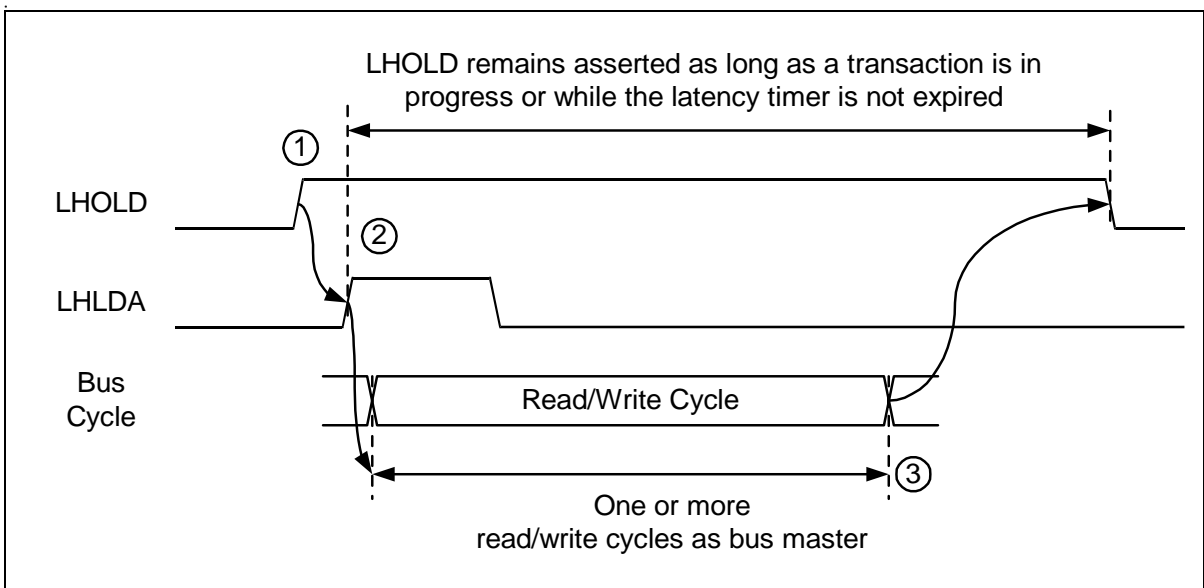


Figure 5-6 Intel Bus Arbitration

Valid C/ $\overline{\text{BE}}$ combinations and the correspondence between local address, $\overline{\text{LBHE}}$ and the mapping of PCI data to the local data bus are shown in table 5-2 and table 5-3. All

Interface Description

accesses not shown in the table result in generation of a 'PCI Access Error' interrupt vector.

Table 5-2 C/ $\overline{\text{BE}}$ to LA/ $\overline{\text{LBHE}}$ mapping in Intel bus mode (8 bit port mode)

| C/ $\overline{\text{BE}}$ (3:0) | LA(1:0) | $\overline{\text{LBHE}}$ | LD(15:8) | LD(7:0) |
|---------------------------------|-----------------|--------------------------|----------|-----------|
| 1110 _B | 00 _B | 1 | - | AD(7:0) |
| 1101 _B | 01 _B | 1 | - | AD(15:8) |
| 1011 _B | 10 _B | 1 | - | AD(23:16) |
| 0111 _B | 11 _B | 1 | - | AD(31:24) |

Table 5-3 C/ $\overline{\text{BE}}$ to LA/ $\overline{\text{LBHE}}$ mapping in Intel bus mode (16 bit port mode)

| C/ $\overline{\text{BE}}$ (3:0) | LA(1:0) | $\overline{\text{LBHE}}$ | LD(15:8) | LD(7:0) |
|---------------------------------|-----------------|--------------------------|-----------|-----------|
| 1110 _B | 00 _B | 1 | - | AD(7:0) |
| 1101 _B | 01 _B | 0 | AD(15:8) | - |
| 1011 _B | 10 _B | 1 | - | AD(23:16) |
| 0111 _B | 11 _B | 0 | AD(31:24) | - |
| 1100 _B | 00 _B | 0 | AD(15:8) | AD(7:0) |
| 0011 _B | 10 _B | 0 | AD(31:24) | AD(23:16) |

5.3.2 Motorola Mode

5.3.2.1 Slave Mode

The demultiplexed bus modes use the local bus port pins LA(12:1) for the 16-bit address and the local bus port pins LD(15:0) for 16-bit data. A read/write access is initiated by placing an address on the address bus and asserting $\overline{LCS0}$ together with the command signal \overline{LWRRD} (see "Motorola Bus Mode" on page 156). The data cycle begins when the signal \overline{LDS} is asserted. Data is driven onto the data bus either by the MUNICH256FM (for read cycles) or by the external processor (for write cycles). After a period of time, which is determined by the access time to the internal registers valid data is placed on the bus, which is indicated by asserting the active low signal \overline{LDTACK} .

Note: $\overline{LCS0}$ need not be deasserted between two subsequent cycles to the same device.

Read cycles

Input data can be latched and the data strobe signal can be deactivated now. This causes the MUNICH256FM to remove its data from the data bus which is then tri-stated again. \overline{LDTACK} is driven high and will be tri-stated as soon as $\overline{LCS0}$ is deasserted.

Write cycles

The data strobe signal can be deactivated now. If a subsequent bus cycle is required, the external processor can place the respective address on the address bus.

5.3.2.2 Master Mode

As in Intel mode a read/write access from the PCI bus to the 16-bit demultiplexed local bus is initiated by accessing the PCI memory space base mapped by the base address register 2. Each valid read or write access to this base address triggers the local bus master interface which in turn starts arbitration for the local bus using the interface signals \overline{LBR} and \overline{LBG} and \overline{LBGACK} . As soon as the MUNICH256FM gets access to the local bus it places a valid address on the address bus, sets the $LSIZE0$ signal which indicates a 8- or 16-bit bus access and asserts the corresponding chip select signal. The signal \overline{LWRRD} indicates a read or write operation. The data cycle begins when the signal \overline{LDS} is asserted. Data is driven onto the data bus either by the MUNICH256FM or by the external component.

A transaction is finished on the local bus when the external device asserts the active low signal \overline{LDTACK} or when the internal wait state timer expires.

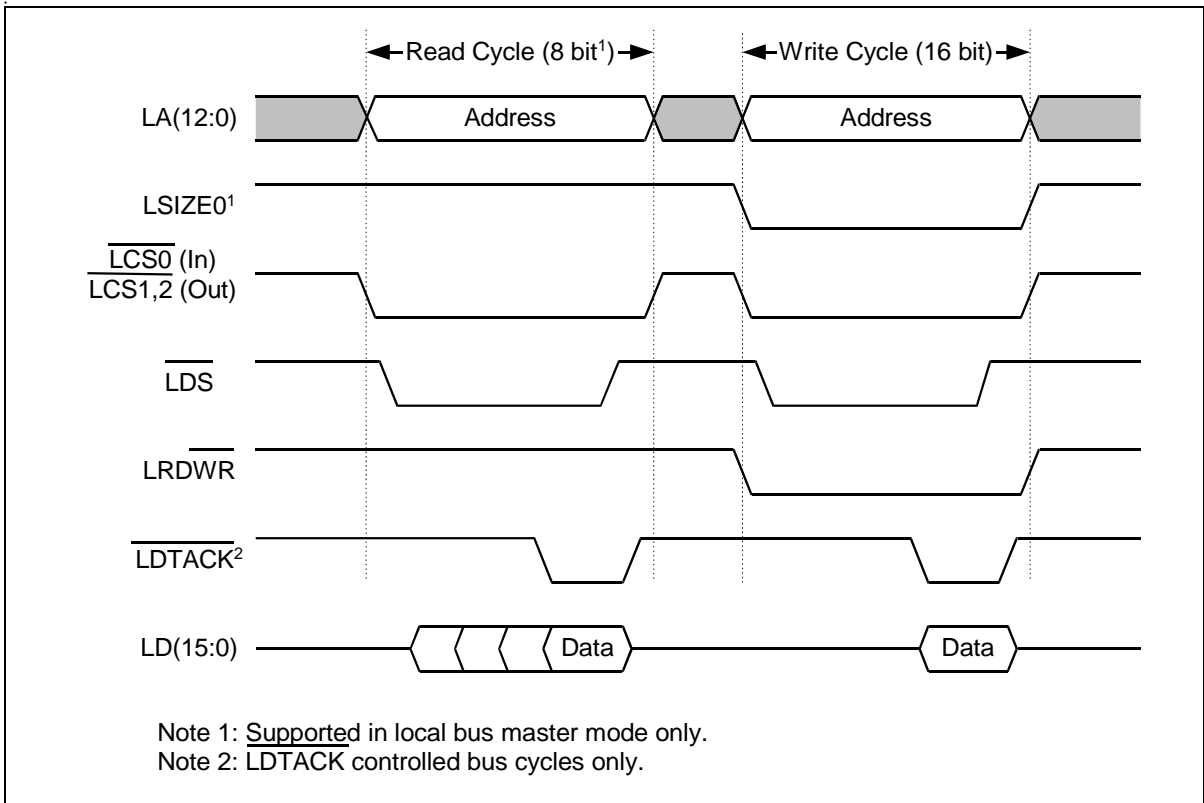


Figure 5-7 Motorola Bus Mode

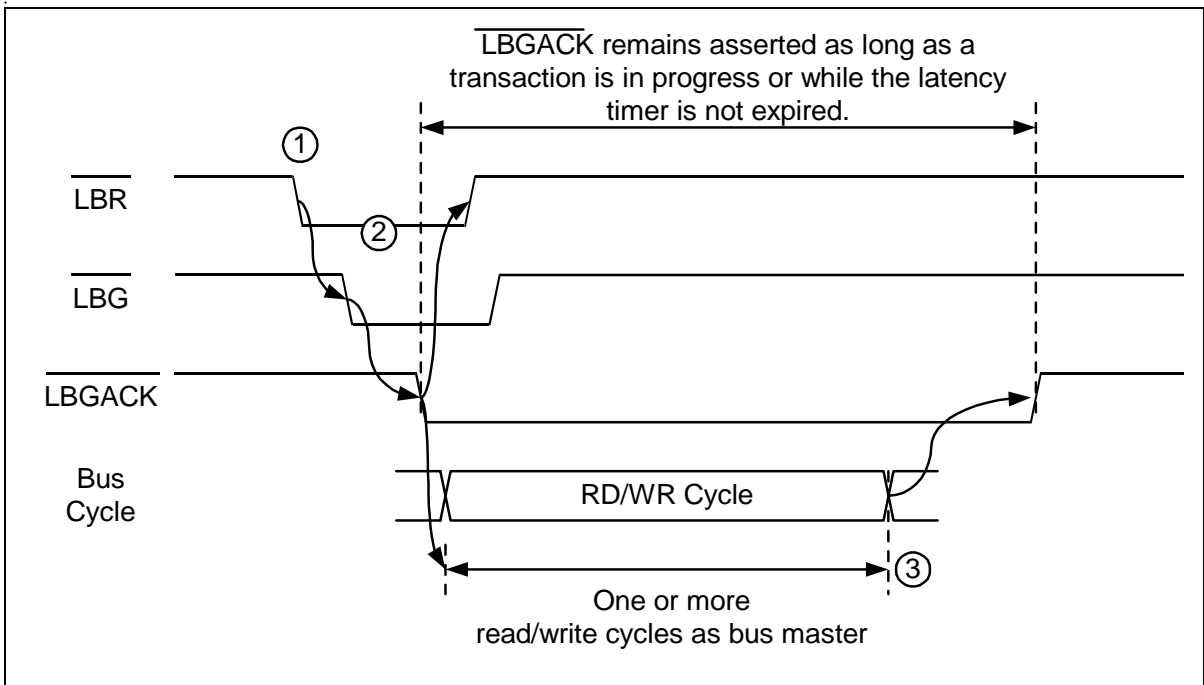


Figure 5-8 Motorola Bus Arbitration

Interface Description

The address and byte enable signals on the PCI bus are mapped to the local bus according to table 5-4 and table 5-5. It can be seen that the MUNICH256FM supports different valid $\overline{C/\overline{BE}}$ combinations which result in either a 8- or 16-bit access to the local bus interface. All accesses not shown in the table result in generation of a 'PCI Access Error' interrupt vector. Byte swapping for 16 bit data transfers can be disabled.

Table 5-4 $\overline{C/\overline{BE}}$ to LA/LSIZE0 mapping in Motorola bus mode (8 bit port mode)

| $\overline{C/\overline{BE}}$ (3:0) | LA(1:0) | LSIZE0 | LD(15:8) | LD(7:0) |
|------------------------------------|-----------------|--------|-----------|---------|
| 1110 _B | 00 _B | 1 | AD(7:0) | - |
| 1101 _B | 01 _B | 1 | AD(15:8) | - |
| 1011 _B | 10 _B | 1 | AD(23:16) | - |
| 0111 _B | 11 _B | 1 | AD(31:24) | - |

Table 5-5 $\overline{C/\overline{BE}}$ to LA/LSIZE0 mapping in Motorola bus mode (16 bit port mode)

| $\overline{C/\overline{BE}}$ (3:0) | LA(1:0) | LSIZE0 | LD(15:8) | LD(7:0) |
|------------------------------------|-----------------|--------|-----------|-----------|
| 1110 _B | 00 _B | 1 | AD(7:0) | |
| 1101 _B | 01 _B | 1 | - | AD(15:8) |
| 1011 _B | 10 _B | 1 | AD(23:16) | - |
| 0111 _B | 11 _B | 1 | - | AD(31:24) |
| 1100 _B | 00 _B | 0 | AD(7:0) | AD(15:8) |
| 0011 _B | 10 _B | 0 | AD(23:16) | AD(31:24) |

5.4 Serial Line Interface

The DS3 interface of the MUNICH256FM consists of one receive port and one transmit port. The receive port provides a clock input (RC44) and one (RD44) or two data inputs (RD44P, RD44N) for unipolar or dual-rail input signals. Receive data can be sampled on the rising or falling edge of the receive clock. In transmit direction the port interface consists of two clock signals, the transmit clock input TC44 and a clock output signal TC44O. The data signals consists of one (TD44) or two data outputs (TD44P, TD44N) for unipolar or dual-rail output signals. The transmit port can be clocked by the receive clock RC44 or by the transmit clock TC44. The selected clock is provided as an output on TC44O. Transmit data is updated on the rising or falling edge of TC44O.

The MUNICH256FM provides two additional serial interfaces, one for DS3 overhead bit access and one for DS3 stuff bit access (M13 asynchronous format only).

The overhead access is provided via an overhead clock signal (ROVHCK, TOVHCK), an overhead data signal (ROVHD, TOVHD) and an synchronization signal (ROVHSYN, TOVHSYN) which marks the X overhead bit of the first subframe of a DS3 signal. In transmit direction the overhead enable signal (TOVHEN) marks those bits which shall be inserted in the overhead bits of the DS3 signal. All overhead signals are updated or sampled on the rising edge of the corresponding overhead clock, i.e. ROVHCK or TOVHCK. See **Figure 5-9** and **Figure 5-10** for details.

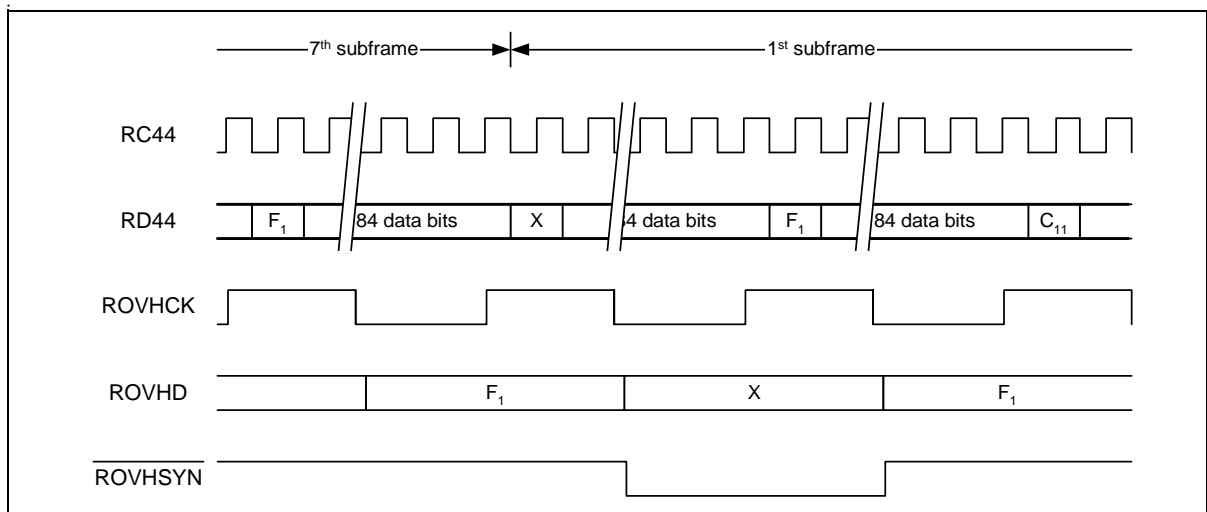
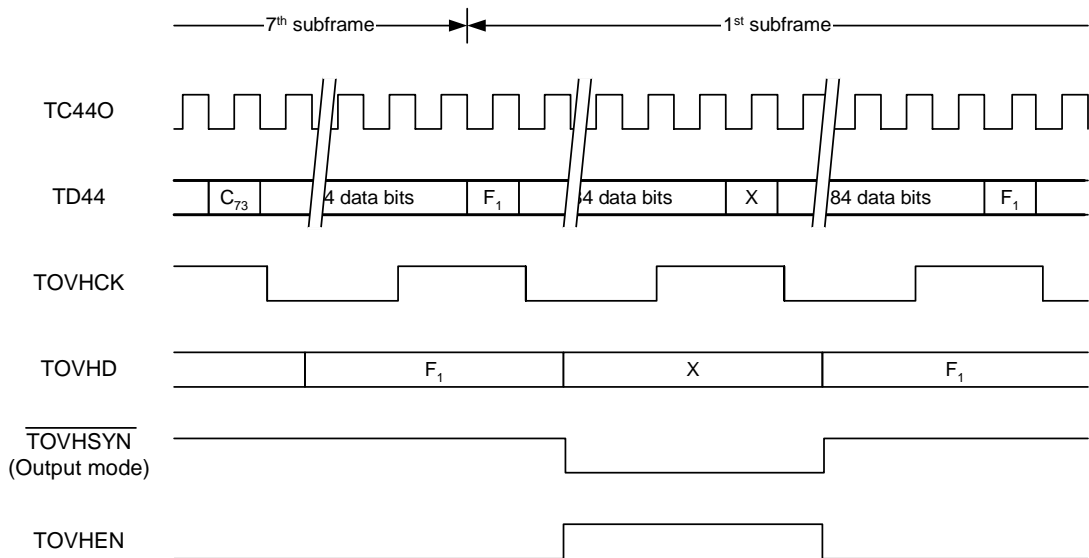


Figure 5-9 Receive Overhead Access

1. Transmit Overhead Bit Access ($\overline{\text{TOVHSYN}}$ in output mode)



2. Transmit Overhead Bit Access ($\overline{\text{TOVHSYN}}$ in input mode)

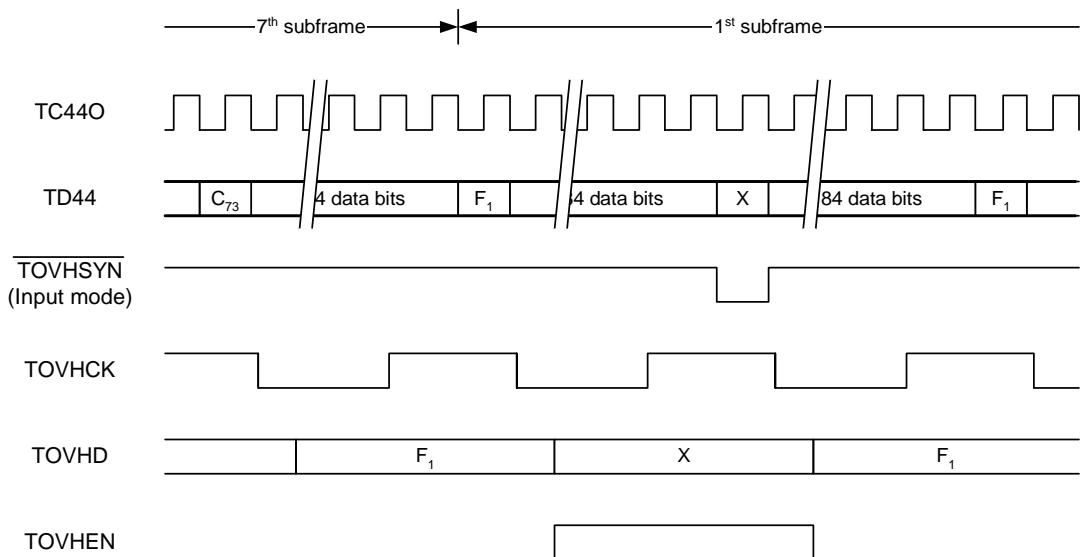


Figure 5-10 Transmit Overhead Access

The stuff bit access is provided via a receive and transmit stuff bit clock (RSBCK, TSBCK) and the two stuff bit signals RSD and TSD. Stuff bits are updated and sampled on the rising edge of the of stuff bit clock.

5.5 JTAG Interface

A test access port (TAP) is implemented in the MUNICH256FM. The essential part of the TAP is a finite state machine (16 states) controlling the different operational modes of the boundary scan. Both, TAP controller and boundary scan, meet the requirements given by the JTAG standard: IEEE 1149.1. **Figure 5-11** gives an overview about the TAP controller.

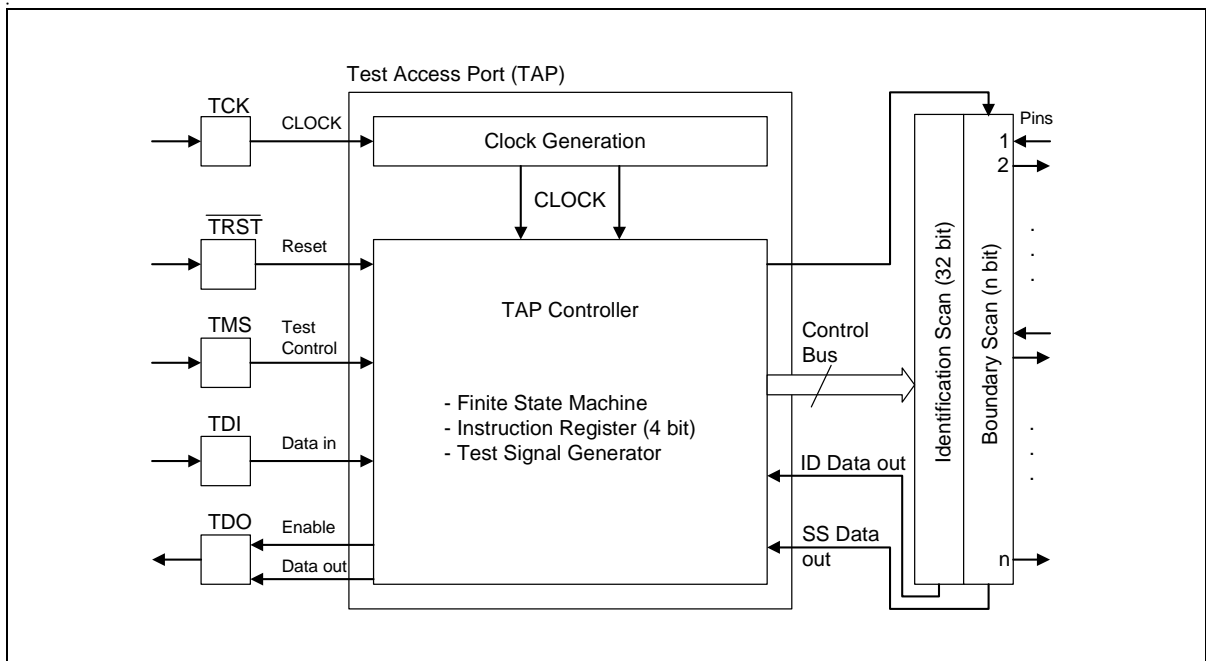


Figure 5-11 Block Diagram of Test Access Port and Boundary Scan Unit

If no boundary scan operation is planned $\overline{\text{TRST}}$ has to be connected with V_{SS} . TMS and TDI do not need to be connected since pull-up transistors ensure high input levels in this case. Nevertheless it would be a good practice to put the unused inputs to defined levels. In this case, if the JTAG is not used:

TMS = TCK = '1' is recommended.

Test handling (boundary scan operation) is performed via the pins TCK (Test Clock), TMS (Test Mode Select), TDI (Test Data Input) and TDO (Test Data Output) when the TAP controller is not in its reset state, i. e. $\overline{\text{TRST}}$ is connected to V_{DD3} or it remains unconnected due to its internal pull up. Test data at TDI are loaded with a clock signal connected to TCK. '1' or '0' on TMS causes a transition from one controller state to another; constant '1' on TMS leads to normal operation of the chip.

An input pin (I) uses one boundary scan cell (data in), an output pin (O) uses two cells (data out, enable) and an I/O-pin (I/O) uses three cells (data in, data out, enable). Note that most functional output and input pins of the MUNICH256FM are tested as I/O pins in boundary scan, hence using three cells. The boundary scan unit of the

Interface Description

MUNICH256FM contains a total of $n = 484$ scan cells. The desired test mode is selected by serially loading a 4-bit instruction code into the instruction register via TDI (LSB first).

EXTEST is used to examine the interconnection of the devices on the board. In this test mode at first all input pins capture the current level on the corresponding external interconnection line, whereas all output pins are held at constant values ('0' or '1'). Then the contents of the boundary scan is shifted to TDO. At the same time the next scan vector is loaded from TDI. Subsequently all output pins are updated according to the new boundary scan contents and all input pins again capture the current external level afterwards, and so on.

INTEST supports internal testing of the chip, i. e. the output pins capture the current level on the corresponding internal line whereas all input pins are held on constant values ('0' or '1'). The resulting boundary scan vector is shifted to TDO. The next test vector is serially loaded via TDI. Then all input pins are updated for the following test cycle.

SAMPLE/PRELOAD is a test mode which provides a snapshot of pin levels during normal operation.

IDCODE: A 32-bit identification register is serially read out via TDO. It contains the version number (4 bits), the device code (16 bits) and the manufacturer code (11 bits). The LSB is fixed to '1'.

The ID code field is set to

Version : 2_H

Part Number : 0076_H

Manufacturer : 083_H (including LSB, which is fixed to '1')

Note: Since in test logic reset state the code '0011' is automatically loaded into the instruction register, the ID code can easily be read out in shift DR state.

BYPASS: A bit entering TDI is shifted to TDO after one TCK clock cycle.

CLAMP allows the state of signals driven from component pins to be determined from the boundary-scan register while the bypass register is selected as the serial path between TDI and TDO. Signals driven from the MUNICH256FM will not change while the CLAMP instruction is selected.

HIGHZ places all of the system outputs in an inactive drive state.

6 Channel Programming / Reprogramming Concept

For channel programming the MUNICH256FM provides a on-chip channel specification data structure. All information necessary to setup a channel has to be provided using this data structure. As soon as all channel information has been written to the channel specification registers the information can be released using simple channel commands, which have to be written to register CSPEC_CMD. The relevant channel information will then be copied to the chip internal channel database. The channel specification registers, which need to be programmed before a command can be executed, are shown in **Table 6-1**.

Before initializing a channel the time slot assignment process for the affected channel must be completed. Vice versa after shutting down a channel the time slots associated with the affected channel should be set to inhibit. Otherwise if a time slot is reprogrammed afterwards, strange behavior can be expected on the serial side.

For each channel a simple sequence of channel commands must be ensured. After reset each channel is in its 'off' state. Therefore, the first command to start a channel is 'Transmit Init' or 'Receive Init'. This brings the channel into the operational state. In this state all commands except 'Transmit Init', 'Receive Init' or 'Transmit Idle' can be given. To bring a channel back into the idle state a 'Transmit Off' or 'Receive Off' command has to be programmed. For certain channel commands system software has to wait before new commands can be given for the same channel. This is due to internal buffer allocation functions which require some processing time. Notification of system software is done in form of command interrupt vectors, which signal that a command has successful or even unsuccessful completed.

Table 6-1 Channel Specification Registers and Channel Commands

| Register | Transmit Commands | | | | | | | Receive Commands | | | | |
|-----------------|-------------------|--------------|-----------------------|---------------------|---------------|----------------|----------------------|------------------|-------------|----------------------|--------------------|---------------|
| | Transmit Init | Transmit Off | Transmit Abort/Branch | Transmit Hold Reset | Transmit Idle | Transmit Debug | Transmit Update FNUM | Receive Init | Receive Off | Receive Abort/Branch | Receive Hold Reset | Receive Debug |
| CSPEC_MODE_REC | | | | | | | | | | | | |
| CSPEC_REC_ACCM | | | | | | | | | | | | |
| CSPEC_MODE_XMIT | | | | | | | | | | | | |

Channel Programming / Reprogramming Concept

| Register | Transmit Commands | | | | | | | Receive Commands | | | | |
|-----------------|-------------------|--------------|-----------------------|---------------------|---------------|----------------|----------------------|------------------|-------------|----------------------|--------------------|---------------|
| | Transmit Init | Transmit Off | Transmit Abort/Branch | Transmit Hold Reset | Transmit Idle | Transmit Debug | Transmit Update FNUM | Receive Init | Receive Off | Receive Abort/Branch | Receive Hold Reset | Receive Debug |
| CSPEC_XMIT_ACCM | | | | | | | | | | | | |
| CSPEC_BUFFER | | | | | | | | | | | | |
| CSPEC_FRDA | | | | | | | | | | | | |
| CSPEC_FTDA | | | | | | | | | | | | |
| CSPEC_IMASK | | | | | | | | | | | | |

6.1 Channel Commands

The following section describes all receive and transmit channel commands and the programming sequence in details.

6.2 Transmit Channel Commands

Transmit Init

Before a 'Transmit Init' command is given, the MUNICH256FM will not transmit data for a channel. After the 'Transmit Init' command the channel database of the affected channel is initialized according to the parameters in the channel specification registers.

After initialization the transmit buffer prepares the buffer locations for the selected channel and the data management unit starts processing the linked list and fills the prepared buffer locations. In order to prevent a transmit underrun condition, the transmit buffer is filled up to the transmit forward threshold before data is sent to the serial side. The protocol machine formats data according to the given channel parameters and the data is placed in the time slots assigned to the selected channel. When no or not sufficient data is available, the device sends the idle code according the selected protocol mode.

If the command was successful, a 'Transmit Command Complete' interrupt vector is generated after the first transmit descriptor is read pointed to by register CSPEC_FTDA. In case that there is insufficient transmit buffer space, the command cannot be

Channel Programming / Reprogramming Concept

completed internally and the device responds with a 'Transmit Command Failed' interrupt vector. Furthermore the MUNICH256FM will not start processing the linked list for this particular channel.

New commands for the same channel may be given after the user received the 'Transmit Command Complete' interrupt vector. Prior to new initialization of the same channel it must be turned off using the 'Transmit Off' command.

Transmit Off

After 'Transmit Off' the transmit channel is disabled immediately and the time slots assigned to the selected channel are set to '1'. The transmit buffer releases all buffer locations assigned to the channel. The data management unit updates the last processed descriptor with the complete bit if enabled and generates a 'Transmit Host Initiated' interrupt vector if the THI bit in the last descriptor was set. All channel related informations are cleared from the internal channel database.

A 'Transmit Command Complete' interrupt vector is generated when the channel command is finished. After that time processing of the linked list is completely stopped. New commands for the same channel may be given after the user received the 'Transmit Command Complete' interrupt vector.

Transmit Abort/Branch

The 'Transmit Abort/Branch' command is performed on the serial side and in the data management unit. The data management unit stops immediately processing the current descriptor and branches to a new descriptor pointed to by CSPEC_FTDA. Data which is already stored in the transmit buffer is sent on the serial side. The protocol machine will append an abort sequence if data in transmit buffer was not complete due to 'Transmit Abort/Branch' command. System software is informed about the aborted frame by a 'Transmit Abort' channel interrupt vector. If no data is stored in the transmit buffer this command does not affect the serial side and no 'Transmit Abort' interrupt vector is generated. Data transmission is continued with a new frame when the data management unit branched to the new descriptor list.

A 'Transmit Command Complete' interrupt vector is generated after the management unit released the old descriptor list. New commands for the same channel may be given after the user received the 'Transmit Command Complete' interrupt vector.

Transmit Hold Reset

The 'Transmit Hold Reset' command must be given after system software has set the HOLD bit of a descriptor from '1' to '0'. In case that the MUNICH256FM is in hold condition it reads the descriptor which had its HOLD bit set and tests the HOLD bit of the descriptor. If the HOLD bit is set to '0' the data management unit branches to the next descriptor and continues data transmission. Otherwise the particular channel remains in hold condition.

Channel Programming / Reprogramming Concept

The MUNICH256FM will NOT generate a 'Transmit Command Complete' interrupt vector after this command is programmed.

Transmit Update FNUM

The 'Transmit Update FNUM' command changes the parameter CSPEC_MODE_XMIT.FNUM in the internal channel database, which allows to change dynamically the number of idle flags that are inserted between two frames.

The MUNICH256FM will NOT generate a 'Transmit Command Complete' interrupt vector after this command is programmed.

Transmit Idle

The 'Transmit Idle' command starts the MUNICH256FM to send the value CSPEC_MODE_XMIT.TFLAG in the time slots of the selected channel. This command can only be given if a channel is turned off.

The MUNICH256FM will NOT generate a 'Transmit Command Complete' interrupt vector after this command is programmed.

Transmit Debug

The 'Transmit Debug' command allows to read back the current settings of the internal channel database. After the 'Transmit Debug' command has been programmed system software can read back the current values of the channel specification registers. Register CSPEC_FTDA contains the value of the next transmit descriptor.

The MUNICH256FM will NOT generate a 'Transmit Command Complete' interrupt vector after this command is programmed.

Note: The setting of the internal channel database is not copied into the channel specification registers and therefore the values read can not be used to program another channel. After system software has used the 'Transmit Debug' command it must reprogram the channel specification registers to setup a new channel.

6.3 Receive Channel Commands

Receive Init

Before a 'Receive Init' command is given, the MUNICH256FM will not process data for a channel. After the 'Receive Init' command the channel database of the affected channel is initialized according to the parameters programmed in channel specification registers.

After initialization data received in those time slots assigned to the selected channel is processed and stored in the internal receive buffer. The data management unit starts

Channel Programming / Reprogramming Concept

storing this data in the linked list which starts at CSPEC_FRDA. The protocol machine deformats and checks data according to the given channel parameters.

A 'Receive Command Complete' interrupt vector is generated after the channel information is copied into the internal channel database.

New commands for the same channel may be given after the MUNICH256FM issued the 'Receive Command Complete' interrupt vector. Prior to new initialization of the same channel it must be turned off using the 'Receive Off' command.

Receive Off

The 'Receive Off' command disables the receive channel immediately. Further incoming data is discarded until the next 'Receive Init' command is given. Data already stored in the receive buffer is written to system memory. If a frame is destroyed by the 'Receive Off' command a 'Receive Abort' channel interrupt vector is generated.

A 'Receive Command Complete' interrupt vector is generated after remaining data in the receive buffer is written to system memory. After that time processing of the linked list is stopped and the channel information is cleared from the internal channel database.

New commands for the same channel may be given after the MUNICH256FM issued the 'Receive Command Complete' interrupt vector.

Receive Abort/Branch

The 'Receive Abort/Branch' command is performed in the data management unit. The data management unit stops immediately processing the current descriptor and branches to a new descriptor pointed to by CSPEC_FRDA. In case that the 'Receive Abort/Branch' command is issued while a packet is written to system memory a 'Receive Abort' interrupt vector is generated and the rest of the frame already stored in receive buffer is discarded. Data reception is continued with a new frame when the data management unit branched to the new descriptor list.

A 'Receive Command Complete' interrupt vector is generated after the channel information is copied into the internal channel database. New commands for the same channel may be given after the MUNICH256FM issued the 'Receive Command Complete' interrupt vector.

Receive Hold Reset

The 'Receive Hold Reset' command must be given after system software has set the HOLD bit of a receive descriptor from '1' to '0'. In case that the MUNICH256FM is in hold condition it reads the descriptor which had its HOLD bit set and tests the HOLD bit of the descriptor. If the HOLD bit is set to '0' the data management unit branches to the next descriptor and continues data reception. Otherwise the particular channel remains in hold condition.

Channel Programming / Reprogramming Concept

The MUNICH256FM will NOT generate a 'Receive Command Complete' interrupt vector after this command is programmed.

Receive Debug

The 'Receive Debug' command allows to read back the current settings of the internal channel database. After the 'Receive Debug' command has been programmed system software can read back the current values of the channel specification registers. Register CSPEC_FRDA contains the value of the next receive descriptor.

The MUNICH256FM will NOT generate a 'Receive Command Complete' interrupt vector after this command is programmed.

Note: The setting of the internal channel database is not copied into the channel specification registers and therefore the values read can not be used to program another channel. After system software has used the 'Receive Debug' command it must reprogram the channel specification registers to setup a new channel.

7 Reset and Initialization procedure

Since the term “initialization” can have different meanings, the following definition applies:

Chip Initialization

Generating defined values in all on-chip registers, RAMs (if required), flip-flops etc.

Mode Initialization

Software procedure, that prepares the device to its required operation, i.e. mainly writing on-chip registers to prepare the device for operation in the respective system environment.

Operational programming

Software procedures that setup, maintain and shut down operational modes, i.e. initialize logical channel or maintain framing operations on selected ports.

7.1 Chip Initialization

Hardware reset

The hardware reset \overline{RST} has to be applied to the device. Chip input \overline{TRST} must be activated prior to or while asserting \overline{RST} and should be held asserted as long as the boundary scan operation is not required. System clock must start running during reset. During reset:

- All I/Os and all outputs are tri-state.
- All registers, state machines, flip-flops etc. are set asynchronously to their reset values and all internal modules are set to their initial state.
- All interrupts are masked.
- The register bit CONF1.STOP is set to ‘1’.

After hardware reset (\overline{RST} deasserted) system clock CLK is assumed to be running. Serial clocks must be low/high or running. The PCI and the local bus interface pins go into their idle state. All serial line outputs are tri-state.

The PCI interface becomes active and depending on input pin SPLOAD starts to read subsystem ID/subsystem vendor ID and Memory commands out of external EEPROM via the SPI interface. The serial clock is derived from the PCI clock. As long as this procedure is active, the PCI interface answers all accesses with retry. After the PCI interface has finished its self initialization it can be configured with PCI configuration cycles.

In parallel to PCI self initialization the internal modules start their RAM initialization. As long as the RAM initialization is running the internal modules indicate this condition with

Reset and Initialization procedure

their initialization in progress signal. The register bit CONF1.IIP is the result of all signals. As soon as all internal modules have finished their RAM initialization the register bit CONF1.IIP is deasserted. Software must poll the register bit CONF1.IIP until this bit has been deasserted. Read access to registers other than CONF1 is prohibited and may result in unexpected behavior of the design. Write accesses are not allowed.

Chip initialization is finished when CONF1.IIP is '0'.

Software Reset

Alternately the MUNICH256FM provides the capability to issue a software reset via register bit CONF1.SRST. During software reset all interfaces except PCI interface are forced into their idle state. After software reset is set the MUNICH256FM starts its self initialization and IIP will be asserted. Chip initialization is finished when CONF1.IIP is deasserted. Afterwards the software reset bit must be set to '0' to allow further operation.

7.2 Mode Initialization

After chip initialization is finished the system software has to setup the device for the required function.

The system software has to poll bit CONF1.IIP (FCONF.IIP). As soon as CONF1.IIP is deasserted, the system software has to clear bit CONF1.STOP and has to set the general operating modes in register CONF1.

The M13 multiplexer, DS3/DS2 framer mode, T1/E1 framer mode and the DS1/E1 and DS3 port interface has to be programmed. It is assumed, that the DS3 port clock and CTCLK are active. The T1/E1 ports shall be disabled, thus no incoming data is forwarded to the time slot assigner and to the T1/E1 framer.

Transmit direction

The T1/E1s have to be enabled via register XPI.TEN. After the tributaries are enabled, the F-Bit (T1 mode) respectively time slot zero (E1 mode) are generated by the on-chip T1/E1 framer and the signalling controller. To synchronize the first bit of a frame to an external reference the common transmit frame synchronization pulse CTFS can be used (in external timing mode only). After a tributary has been enabled, payload data is provided from the time slot assigner. Since the time slot assignment is in reset state, that is all time slots are set to inhibit, data bits are sent as '1'.

Receive direction

The tributaries have to be enabled via register XPI.REN. After they are enabled, the on-chip T1/E1 framer tries to achieve frame alignment. As soon as frame alignment has been achieved, incoming payload data is passed to the time slot assigner. Since time slot assignment is in reset state, that is all time slots are set to inhibit, data bits are discarded.

8 Register Description

The register description of the MUNICH256FM is divided into two parts, an overview of all internal registers and in the second part a detailed description of all internal registers.

8.1 Register Overview

The first part of the register overview describes the PCI configuration space registers. The second part describes the register set which can be accessed from PCI side only. These registers are used to setup the main operation modes and to run the channel engines of the device. The last part describes the register set of the framing engines, the signalling controller, the mailbox and the local interrupt FIFO. These registers may be accessed through the local microprocessor interface or via PCI.

Note: Register locations not contained in the following register tables are “reserved”. In general all write accesses to reserved registers are discarded and read access to reserved registers result in 00000000_H. Nevertheless, to allow future extensions, system software shall access documented registers only, since writes to reserved registers may result in unexpected behavior. The read value of reserved registers shall be handled as don't care.

Unused and reserved bits are marked with a gray box. The same rules as given for register accesses apply to reserved bits, except that system software shall write the documented default value in reserved bit locations.

8.1.1 PCI Configuration Register Set (Direct Access)

Table 8-1 PCI Configuration Register Set

| Register | Access | Address | Reset value | Comment | Page |
|--|--------|----------------------------------|-----------------------|--|------|
| Standard configuration space register | | | | | |
| DID/VID | R | 00 _H | 2108110A _H | Device ID/Vendor ID | 182 |
| STA/CMD | R/W | 04 _H | 02A00000 _H | Status/Command | 183 |
| CC/RID | R | 08 _H | 02800001 _H | Class Code/Revision ID | 185 |
| BIST/ HEAD/ LATIM/ CLSIZ | R/W | 0C _H | 00000000 _H | Built-in Self Test/ Header Type/ Latency Timer/ Cache Line Size | 186 |
| BAR1 | R/W | 10 _H | 00000000 _H | Base Address 1 | 187 |
| BAR2 | R/W | 14 _H | 00000000 _H | Base Address 2 | 188 |
| BARX | R | 14 _H -24 _H | 00000000 _H | Base Address Not Used | |

Register Description

| Register | Access | Address | Reset value | Comment | Page |
|--|--------|-----------------|-----------------------|--|------|
| CISP | R | 28 _H | 00000000 _H | Cardbus CIS Pointer | |
| SSID/ SSVID | R | 2C _H | 00000000 _H | Subsystem ID/ Subsystem Vendor ID | 189 |
| ERBAD | R | 30 _H | 00000000 _H | Expansion ROM Base Adr. | |
| Reserved | R | 34 _H | 00000000 _H | Reserved | |
| Reserved | R | 38 _H | 00000000 _H | Reserved | |
| MAXLAT/ MINGNT/ INTPIN/ INTLIN | R/W | 3C _H | 06020100 _H | Maximum Latency/ Minimum Grant/ Interrupt Pin/ Interrupt Line | 190 |
| User defined configuration space register | | | | | |
| SPI | R/W | 40 _H | 0000001F _H | SPI Access Register | 191 |
| REQ | R/W | 44 _H | 00000000 _H | REQ/GNT Config Register | 193 |
| MEM | R/W | 48 _H | 000007E6 _H | PCI Memory Command | 194 |
| DEBUG | R | 4C _H | 00000000 _H | PCI Debug Support | 196 |

8.1.2 PCI Slave Register Set (Direct Access)

This section shows all registers which are located on the first configuration bus. These registers are used to setup the basic operating modes of the device and to setup the port, time slots and channels. System software has access to these registers via the PCI bus.

Table 8-2 PCI Slave Register Set

| Register | Access | Address | Reset value | Comment | Page |
|--|--------|------------------|-----------------------|--|------|
| General Control | | | | | |
| CONF1 | R/W | 040 _H | | Configuration Register 1 | 214 |
| CONF2 | R/W | 044 _H | 00000000 _H | Configuration Register 2 | 217 |
| CONF3 | R/W | 048 _H | 00090000 _H | Configuration Register 3 | 219 |
| RBAFT | W | 04C _H | 00000000 _H | Receive Buffer Access Failed Interrupt Threshold | 220 |
| SFDT | W | 050 _H | 00000000 _H | Small Frame Dropped Interrupt Threshold Register | 221 |
| Interrupt control PCI bus side | | | | | |
| IQIA | R/W | 0E0 _H | 00000000 _H | Interrupt Queue Initialization | 238 |
| IQBA | R/W | 0E4 _H | 00000000 _H | Interrupt Queue Base Addr. | 240 |
| IQBL | R/W | 0E8 _H | 00000000 _H | Interrupt Queue Length | 241 |
| IQMASK | R/W | 0EC _H | 00000000 _H | Interrupt Queue Mask | 242 |
| GISTA/GIACK | R/W | 0F0 _H | 00000000 _H | Global Interrupt Status/ Global Interrupt Acknowledge | 243 |
| GMASK | R/W | 0F4 _H | FFFFFFFF _H | Interrupt Mask | 245 |
| Channel specification registers (* = CSPEC) | | | | | |
| *_CMD | W | 000 _H | 00000000 _H | Command | 197 |
| *_MODE_REC | R/W | 004 _H | 00000000 _H | Mode Receive | 199 |
| *_REC_ACCM | R/W | 008 _H | 00000000 _H | Receiver ACCM Map | 202 |
| *_MODE_XMIT | R/W | 014 _H | 00000000 _H | Mode Transmit | 203 |
| *_XMIT_ACCM | R/W | 018 _H | 00000000 _H | Transmit ACCM Map | 206 |
| *_BUFFER | R/W | 020 _H | 00200000 _H | Buffer Configuration | 207 |
| *_FRDA | R/W | 024 _H | 00000000 _H | First Receive Descriptor Addr. | 210 |

Register Description

| Register | Access | Address | Reset value | Comment | Page |
|---|--------|------------------|-----------------------|--------------------------------------|------|
| *_FTDA | R/W | 028 _H | 00000000 _H | First Transmit Descriptor Address | 211 |
| *_IMASK | R/W | 02C _H | 00000000 _H | Interrupt Vector Mask | 212 |
| Port and time slot control registers | | | | | |
| PMIAR | R/W | 060 _H | 00000000 _H | Port Mode Indirect Access | 222 |
| PMR | R/W | 064 _H | 0104C000 _H | Port Mode | 223 |
| REN | R/W | 068 _H | 00000000 _H | Receive Enable | 225 |
| TEN | R/W | 06C _H | 00000000 _H | Transmit Enable | 226 |
| TSAIA | R/W | 070 _H | 00000000 _H | Time slot Assignment Indirect Access | 227 |
| TSAD | R/W | 074 _H | 02000000 _H | Time slot Assignment Data | 229 |
| PPP character map/ demap registers | | | | | |
| REC_ACCMX | R/W | 080 _H | 00000000 _H | Receive Extended ACCM Map | 231 |
| XMIT_ACCMX | R/W | 090 _H | 00000000 | Transmit Extended ACCM Map | 235 |
| Receive buffer control | | | | | |
| RBMON | R | 0B0 _H | 02000BFF _H | Receive Buffer Monitor | 236 |
| RBTH | R/W | 0B4 _H | 02000001 _H | Receive Buffer Threshold Report | 237 |
| Maintenance | | | | | |
| RBAFC | R | 084 _H | 00000000 _H | Receive Buffer Access Failed Counter | 232 |
| SFDIA | R/W | 088 _H | 00000000 _H | Small Frame Dropped Indirect Access | 233 |
| SFDC | R | 08C _H | 00000000 _H | Small Frame Dropped Counter | 234 |

8.1.3 PCI and Local Bus Register Set (Direct Access)

This section describes the registers which are located on the configuration bus II (see also "MUNICH256FM Block Diagram" on page 42). These registers can be accessed either from PCI bus via the internal bus bridge or from the local bus side.

Note: Since the local bus is 16-bit wide and the PCI bus is 32-bit wide, the upper 16 bit of data coming from/to PCI are discarded.

Note: Please note that read accesses to local bus registers via PCI bus and therefore the internal bus bridge may result in latencies which exceed the 16 clock rule of PCI specification. Exceeding the 16 clock rule results in target initiated retry on PCI bus. In this case the read cycle needs to be repeated.

Table 8-3 PCI and Local Bus Slave Register Set

| Register | Access | Address (PCI) | Address (Local Bus) | Reset value | Comment | Page |
|--|--------|------------------|---------------------|-------------------|------------------------------------|------|
| FCONF | R/W | 100 _H | 00 _H | 8080 _H | Configuration Register | 246 |
| MTIMER | R/W | 100 _H | 00 _H | 0001 _H | Master Local Bus Timer | 248 |
| Interrupt control for local bus side | | | | | | |
| INTCTRL | R/W | 108 _H | 04 _H | 0001 _H | Interrupt Control | 249 |
| INTFIFO | R | 10C _H | 06 _H | FFFF _H | Interrupt FIFO | 250 |
| DS3 Clock Configuration and Status Register | | | | | | |
| D3CLKCS | R/W | 180 _H | 40 _H | 0000 _H | DS3 Clock Configuration and Status | 262 |
| TUCLKC | R/W | 184 _H | 42 _H | 0000 _H | Test Unit Clock Configuration | 264 |
| DS3 Transmit Control Registers | | | | | | |
| D3TCFG | R/W | 188 _H | 44 _H | 0000 _H | Transmit Configuration | 265 |
| D3TCOM | R/W | 18C _H | 46 _H | 0070 _H | Transmit Command | 267 |
| D3TLPB | R/W | 190 _H | 48 _H | 0000 _H | Remote DS2 Loopback | 269 |
| D3TLPC | R/W | 194 _H | 4A _H | 0000 _H | Transmit Loopback Code Insertion | 270 |
| D3TAIS | R/W | 198 _H | 4C _H | 0000 _H | Transmit AIS Insertion | 271 |
| D3TFINS | R/W | 19C _H | 4E _H | 0000 _H | Transmit Fault Insertion Control | 272 |

Register Description

| Register | Access | Address (PCI) | Address (Local Bus) | Reset value | Comment | Page |
|---|--------|------------------|---------------------|-------------------|---|------|
| D3TTUC | R/W | 1A0 _H | 50 _H | 0000 _H | Transmit Test Unit Control | 273 |
| D3TSDL | R/W | 1A4 _H | 52 _H | 01FF _H | Transmit Spare Data Link | 274 |
| DS3 Receive Control/Status Registers | | | | | | |
| D3RCFG | R/W | 1C0 _H | 60 _H | 0000 _H | Receive Configuration | 275 |
| D3RCOM | R/W | 1C4 _H | 62 _H | 0000 _H | Receive Command | 278 |
| D3RIMSK | R/W | 1C8 _H | 64 _H | 1FFF _H | Receive Interrupt Mask | 280 |
| D3RESIM | R/W | 1CC _H | 66 _H | 0000 _H | Receive Error Simulation | 281 |
| D3RTUC | R/W | 1D0 _H | 68 _H | 0000 _H | Receive Test Unit Control | 282 |
| D3RSTAT | R | 1D4 _H | 6A _H | 0841 _H | Receive Status | 283 |
| D3RLPCS | R | 1D8 _H | 6C _H | 0000 _H | Receive Loopback Code Status | 286 |
| D3RSDL | R | 1DC _H | 6E _H | 01FF _H | Receive Spare Data Link | 287 |
| D3RCVE | R/W | 1E0 _H | 70 _H | 0000 _H | Receive B3ZS Code Violation Error Counter | 288 |
| D3RFEC | R/W | 1E4 _H | 72 _H | 0000 _H | Receive Framing Bit Error Counter | 288 |
| D3RPEC | R/W | 1E8 _H | 74 _H | 0000 _H | Receive Parity Bit Error Counter | 289 |
| D3RCPEC | R/W | 1EC _H | 76 _H | 0000 _H | Receive CP-Bit Error Counter | 289 |
| D3RFEBEC | R/W | 1F0 _H | 78 _H | 0000 _H | Receive FEBE Error Counter | 290 |
| D3REXZ | R/W | 1F4 _H | 7A _H | 0000 _H | Receive Exzessive Zero Counter | 290 |
| D3RAP | R/W | 1F8 _H | 7C _H | 0000 _H | Alarm Timer Parameter | 291 |

Register Description

| Register | Access | Address (PCI) | Address (Local Bus) | Reset value | Comment | Page |
|---------------------------------------|--------|------------------|---------------------|-------------------|-----------------------------------|------|
| DS2 Transmit Control Registers | | | | | | |
| D2TSEL | R/W | 200 _H | 80 _H | 0000 _H | DS2 Transmit Group Select | 292 |
| D2TCFG | R/W | 204 _H | 82 _H | 0000 _H | Transmit Configuration | 293 |
| D2TCOM | R/W | 208 _H | 84 _H | 0000 _H | Transmit Command | 294 |
| D2TLPC | R/W | 20C _H | 86 _H | 0000 _H | Transmit Loopback Code Insertion | 295 |
| DS2 Receive Control Registers | | | | | | |
| D2RSEL | R/W | 220 _H | 90 _H | 0000 _H | DS2 Receive Group Select | 296 |
| D2RCFG | R/W | 224 _H | 92 _H | 0000 _H | Receive Configuration | 297 |
| D2RCOM | R/W | 228 _H | 94 _H | 0000 _H | Receive Command | 298 |
| D2RIMSK | R/W | 22C _H | 96 _H | 003F _H | Receive Interrupt Mask | 300 |
| D2RSTAT | R | 230 _H | 98 _H | 0001 _H | Receive Status | 301 |
| D2RLPCS | RD | 234 _H | 9A _H | 0000 _H | Receive Loopback Code Status | 303 |
| D2RFEC | R/W | 238 _H | 9C _H | 0000 _H | Receive Framing Bit Error Counter | 304 |
| D2RPEC | R/W | 23C _H | 9E _H | 0000 _H | Receive Parity Bit Error Counter | 304 |
| D2RAP | R/W | 240 _H | A0 _H | 0000 _H | Alarm Timer Parameter | 305 |
| Test Unit Transmit Registers | | | | | | |
| TUTCFG | R/W | 280 _H | C0 _H | 0000 _H | Transmit Configuration | 307 |
| TUTCOM | W | 284 _H | C2 _H | 0000 _H | Transmit Command | 308 |
| TUTEIR | R/W | 288 _H | C4 _H | 0000 _H | Transmit Error Insertion Rate | 310 |
| TUTFP0 | R/W | 28C _H | C6 _H | 0000 _H | Transmit Fixed Pattern | 311 |
| TUTFP1 | R/W | 290 _H | C8 _H | 0000 _H | | |
| Test Unit Receive Registers | | | | | | |
| TURCFG | R/W | 2A0 _H | D0 _H | 0000 _H | Receive Configuration | 312 |

Register Description

| Register | Access | Address (PCI) | Address (Local Bus) | Reset value | Comment | Page |
|--|--------|------------------|---------------------|-------------------|--|------|
| TURCOM | W | 2A4 _H | D2 _H | 0000 _H | Receive Command | 314 |
| TURERMI | R/W | 2A8 _H | D4 _H | 0000 _H | Receive Error Rate Measurement Interval | 316 |
| TURIMSK | R/W | 2AC _H | D6 _H | 001F _H | Receive Interrupt Mask | 317 |
| TURSTAT | R | 2B0 _H | D8 _H | 0021 _H | Receive Status | 318 |
| TURBC0 | R | 2B4 _H | DA _H | 0000 _H | Receive Bit Counter | 320 |
| TURBC1 | R | 2B8 _H | DC _H | 0000 _H | | |
| TUREC0 | R | 2BC _H | DE _H | 0000 _H | Receive Error Counter | 322 |
| TUREC1 | R | 2C0 _H | E0 _H | 0000 _H | | |
| TURFP0 | R | 2C4 _H | E2 _H | 0000 _H | Receive Fixed Pattern | 324 |
| TURFP1 | R | 2C8 _H | E4 _H | 0000 _H | | |
| T1/E1 Framer transmit registers | | | | | | |
| TREGSEL | R/W | 110 _H | 08 _H | 0000 _H | Transmit T1/E1 Framer Port & Register Select | 251 |
| TDATA | R/W | 114 _H | 0A _H | 0000 _H | Transmit T1/E1 Framer Data | 252 |
| T1/E1 Framer receive registers | | | | | | |
| RREGSEL | R/W | 118 _H | 0C _H | 0000 _H | Receive T1/E1 Framer Port & Register Select | 253 |
| RDATA | R/W | 11C _H | 0E _H | 0000 _H | Receive T1/E1 Framer Data | 254 |
| Facility data link registers | | | | | | |
| FREGSEL | R/W | 120 _H | 10 _H | 0000 _H | Facility Data Link Port & Register Select | 255 |
| FDATA | R/W | 124 _H | 12 _H | 0000 _H | Facility Data Link Data | 257 |
| Mailbox registers | | | | | | |
| MBE2P0 | R/W | 140 _H | 20 _H | 0000 _H | Mailbox Local Bus to PCI Command | 258 |

Register Description

| Register | Access | Address (PCI) | Address (Local Bus) | Reset value | Comment | Page |
|--|--------|--|---|-------------------|---|------|
| MBE2P1 MBE2P2 MBE2P3 MBE2P4 MBE2P5 MBE2P6 MBE2P7 | R/W | 144 _H 148 _H 14C _H 150 _H 154 _H 158 _H 15C _H | 22 _H 24 _H 26 _H 28 _H 2A _H 2C _H 2E _H | 0000 _H | Mailbox Local Bus to PCI Data Registers 1 through 7 | 259 |
| MBP2E0 | R/W | 160 _H | 30 _H | 0000 _H | Mailbox PCI to Local Bus Command | 260 |
| MBP2E1 MBP2E2 MBP2E3 MBP2E4 MBP2E5 MBP2E6 MBP2E7 | R/W | 164 _H 168 _H 16C _H 170 _H 174 _H 178 _H 17C _H | 32 _H 34 _H 36 _H 38 _H 3A _H 3C _H 3E _H | 0000 _H | Mailbox PCI to Local Bus Data Registers 1 through 7 | 261 |

8.1.4 Transmit T1/E1 Framer Registers (Indirect Access)

Note: The transmit framer registers will be accessed via registers TREGSEL and TDATA as part of the Local Bus direct access register set. Please refer to page 251 for description of TREGSEL and to page 252 for description of TDATA.

Table 8-4 Transmit T1/E1 Framer Registers

| Register | Access | Address | Reset value | Comment | Page |
|--------------------------|--------|-----------------|-------------------|-------------------------|------|
| Control registers | | | | | |
| TCMDR | R/W | 00 _H | 0000 _H | Command | 325 |
| TFMR | R/W | 01 _H | 0000 _H | Mode | 327 |
| TLCR0 | R/W | 02 _H | 0000 _H | Loop Code Register 0 | 329 |
| TLCR1 | R/W | 03 _H | 0000 _H | Loop Code Register 0 | 330 |
| TPRBSC | R/W | 04 _H | 001F _H | PRBS Control | 331 |
| TFPR0 | R/W | 05 _H | 0000 _H | Fixed Pattern Register | 332 |
| TFPR1 | R/W | 06 _H | 0000 _H | | |
| TPTSL0 | R/W | 07 _H | FFFF _H | PRBS Time slot Register | 333 |
| TPTSL1 | R/W | 08 _H | FFFF _H | | |
| XSP | R/W | 09 _H | 0000 _H | Spare bit Register | 334 |

Register Description

8.1.5 Receive T1/E1 Framer Registers (Indirect Access)

Note: The receive framer registers will be accessed via the registers RREGSEL and RDATA. Please refer to page 253 for description of RREGSEL and to page 254 for description of RDATA.

Table 8-5 Receive T1/E1 Framer Registers

| Register | Access | Address | Reset value | Comment | Page |
|--------------------------|--------|-----------------|-------------------|-------------------------|------|
| Control Registers | | | | | |
| RCMDR | R/W | 00 _H | 0000 _H | Command | 335 |
| RFMR | R/W | 01 _H | 0000 _H | Mode Register | 338 |
| RLCR0 | R/W | 02 _H | 0000 _H | Loop Code Register 0 | 343 |
| RLCR1 | R/W | 03 _H | 0000 _H | Loop Code Register 1 | 344 |
| RPRBSC | R/W | 04 _H | 001F _H | PRBS Control | 345 |
| PFPR0 | R/W | 05 _H | 0000 _H | Fixed Pattern Register | 346 |
| RFPR1 | R/W | 06 _H | 0000 _H | | |
| RPTSL0 | R/W | 07 _H | FFFF _H | PRBS Time slot Register | 347 |
| RPTSL1 | R/W | 08 _H | FFFF _H | | |
| IMR | R/W | 09 _H | 0000 _H | Interrupt Mask | 348 |
| RFMR1 | R/W | 0A _H | 0000 _H | Mode Register 1 | 349 |
| PCD | R/W | 0B _H | 0015 _H | Pulse Count Detection | 350 |
| PCR | R/W | 0C _H | 0015 _H | Pulse Count Recovery | 351 |
| Status registers | | | | | |
| FRS | R | 40 _H | 0000 _H | Status | 352 |
| FEC | R | 41 _H | 0000 _H | Framing Error Counter | 355 |
| CEC | R | 42 _H | 0000 _H | CRC Error Counter | 356 |
| EBC | R | 43 _H | 0000 _H | Errored Block Counter | 357 |
| BEC | R | 44 _H | 0000 _H | Bit Error Counter | 358 |

8.1.6 Facility Data Link Registers (Indirect Access)

Note: The FDL registers will be accessed via registers FREGSEL and FDATA.

Table 8-6 Facility Data Link Registers

| Register | Access | Address | Reset value | Comment | Page |
|----------|--------|-----------------|-------------------|-----------------------------------|------|
| RCR1 | R/W | 00 _H | 0000 _H | Receive Configuration Register 1 | 359 |
| RCR2 | R/W | 01 _H | 0000 _H | Receive Configuration Register 2 | 362 |
| RFF | R | 02 _H | 0000 _H | Receive FIFO | 364 |
| XCR1 | R/W | 03 _H | 0000 _H | Transmit Configuration Register 1 | 365 |
| XCR2 | R/W | 04 _H | 0000 _H | Transmit Configuration Register 2 | 367 |
| XFF | W | 05 _H | 0000 _H | Transmit FIFO | 368 |
| PSR | R | 06 _H | 0000 _H | Port Status | 369 |
| HND | W | 07 _H | 0000 _H | Handshake | 371 |
| MSK | R/W | 08 _H | 0000 _H | Interrupt Mask | 374 |
| RAL | R/W | 09 _H | 0000 _H | Receive Address Low | 375 |
| RAH | R/W | 0A _H | 0000 _H | Receive Address High | 376 |
| RSAW1 | R | 0B _H | 0000 _H | Receive S _a Word 1 | 377 |
| RSAW2 | R | 0C _H | 0000 _H | Receive S _a Word 2 | 378 |
| RSAW3 | R | 0D _H | 0000 _H | Receive S _a Word 3 | 379 |
| RSAW4 | R | 0E _H | 0000 _H | Receive S _a Word 4 | 380 |
| CRCS1 | R | 0F _H | 0000 _H | CRC Status Counter 1 | 381 |
| CRCS2 | R | 10 _H | 0000 _H | CRC Status Counter 2 | 382 |
| XSAW1 | R/W | 11 _H | 0000 _H | Transmit S _a Word 1 | 383 |
| XSAW2 | R/W | 12 _H | 0000 _H | Transmit S _a Word 2 | 384 |
| XSAW3 | R/W | 13 _H | 0000 _H | Transmit S _a Word 3 | 385 |
| VSSM | R/W | 14 _H | 0000 _H | Valid SSM Pattern | 386 |
| VCRC | R/W | 15 _H | 0000 _H | Valid CRC Count Pattern | 387 |

8.2 Detailed Register Description

8.2.1 PCI Configuration Register

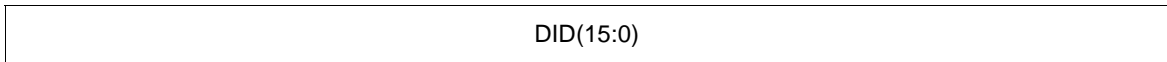
DID/VID

Device ID/Vendor ID

Access : read
Address : 00_H
Reset Value : 2108110A_H

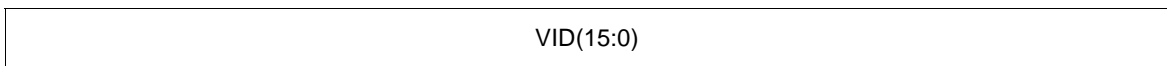
31

16



15

0



DID Device ID
The device ID identifies the particular device. It is hardwired to value 2108_H.

VID Vendor ID
The vendor ID identifies the manufacturer of the device. It is hardwired to value 110A_H.

STAT/CMD
Status/Command Register

Access : read/write
Address : 04_H
Reset Value : 02A00000_H

| | | | | | | | | | | | | | | | | | |
|-----|-----|-----|-----|----|-----------------|----|------|----|-----|----|---|---|---|----|----|----|---|
| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | | | | | | 16 | |
| DPE | SSE | RMA | RTA | 0 | 01 _B | | DPED | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | |
| | | | | | | | 8 | 6 | | | | | | | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | SE | 0 | PER | 0 | 0 | 0 | 0 | BM | MS | 0 | |

- DPE** Detected Parity Error
This bit will be asserted whenever the MUNICH256FM detects a parity error.
0 No parity error detected.
1 Parity error detected. This bit will be cleared by writing a '1' to this bit position.
- SSE** Signaled System Error
This bit will be asserted whenever the MUNICH256FM asserted $\overline{\text{SERR}}$. For system error conditions see bit SE.
0 No system error signaled.
1 System error has been signaled. This bit will be cleared by writing a '1' to this bit position.
- RMA** Received Master Abort
This bit will set whenever a transaction in which the MUNICH256FM acted as bus master was terminated with master abort.
0 No master abort detected.
1 Transaction terminated with master abort. This bit will be cleared by writing a '1' to this bit.

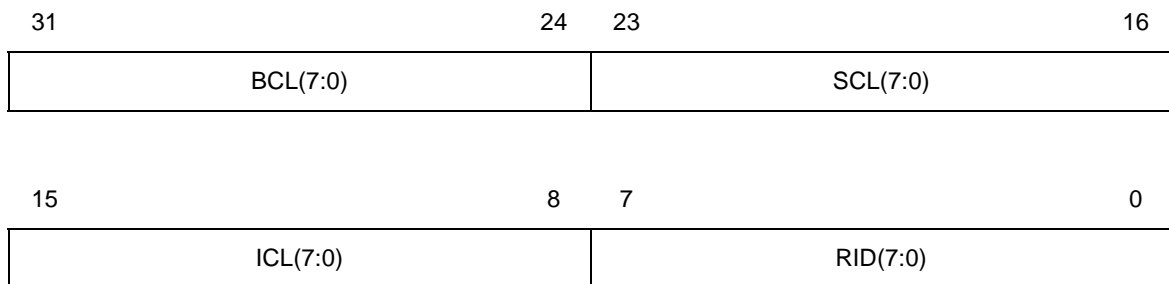
Register Description

| | |
|------|---|
| RTA | <p>Received Target Abort</p> <p>This bit will be set whenever a transaction in which the MUNICH256FM acted as bus master was terminated with target abort.</p> <p>0 No target abort detected.</p> <p>1 Transaction terminated with target abort. This bit will be cleared by writing a '1' to this bit.</p> |
| DPED | <p>Data Parity Error Detected</p> <p>0 No data parity error detected.</p> <p>1 The following three conditions are met:</p> <ul style="list-style-type: none"> •The bus agent asserted $\overline{\text{PERR}}$ itself or observed $\overline{\text{PERR}}$ asserted. •The bus agent acted as bus master for the operation in which the error occurred. •The Parity Error Response Bit is set |
| SE | <p>$\overline{\text{SERR}}$ Enable</p> <p>This bit enables assertion of $\overline{\text{SERR}}$ in case of severe system errors.</p> <p>0 Assertion of $\overline{\text{SERR}}$ disabled.</p> <p>1 Enables report of</p> <ul style="list-style-type: none"> •Address parity errors •Master abort •Target abort |
| PER | <p>Parity Error Response</p> <p>This bit enables reporting of parity errors via pin $\overline{\text{PERR}}$.</p> <p>0 Assertion of $\overline{\text{PERR}}$ disabled.</p> <p>1 Enables the assertion of $\overline{\text{PERR}}$. See also Data Parity Error Detected.</p> |
| BM | <p>Bus Master</p> <p>This bit controls a device ability to act as a master on PCI bus.</p> <p>0 Disables the device from generating PCI accesses.</p> <p>1 Allows the device to act as bus master.</p> |
| MS | <p>Memory Space</p> <p>This bit controls the device response to memory space accesses.</p> <p>0 Response to memory space accesses disabled.</p> <p>1 Allows a device to respond to memory space accesses.</p> |

CC/RID

Class Code/Revision ID

Access : read
Address : 08_H
Reset Value : 02800001_H

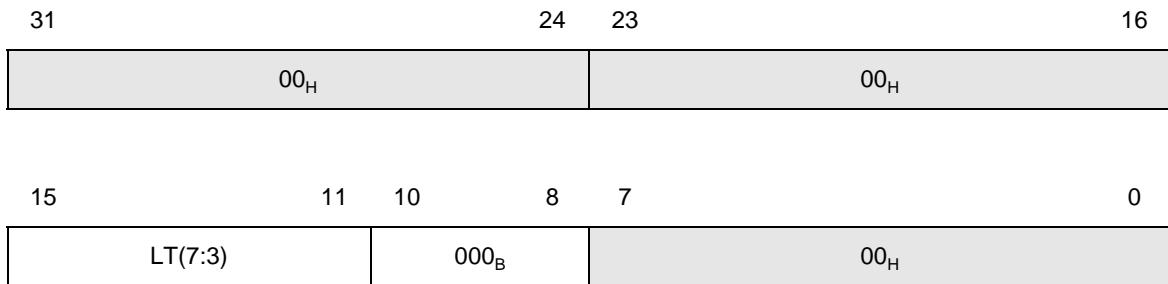


The class code, consisting of base class, subsystem class and interface class, is used to identify the generic function of the device and, in some cases, a specific register-level programming interface.

- BCL** Base Class
The base class is hardwired to 02_H, which identifies this device as a network controller.
- SCL** Sub Class
The sub class is hardwired to 80_H, which together with the base class identifies this device as 'Other network controller'.
- ICL** Interface Class
The interface class is hardwired to 00_H.
- RID** Revision ID
The revision ID identifies the current version of the device. It is hardwired to 01_H.

BIST/Header Type/Latency Timer/Cache Line Size

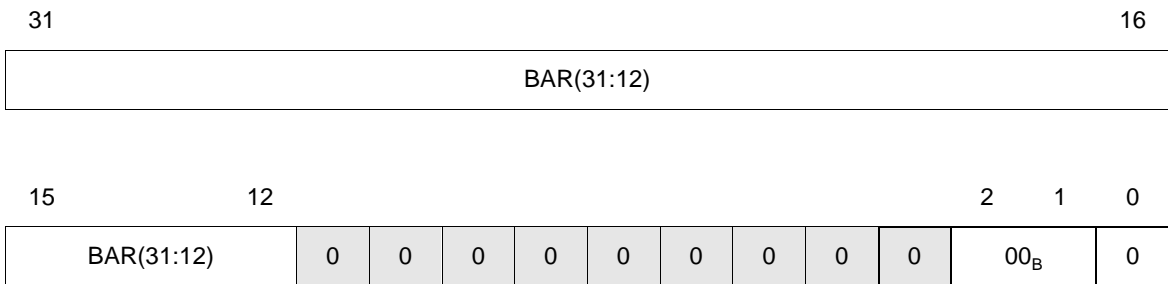
Access : read/write
 Address : 0C_H
 Reset Value : 00000000_H



LT **Latency Timer**
 The value of this register times eight specifies, in units of PCI clocks, the value of the latency timer for this PCI bus master.

BAR1
Base Address 1

Access : read/write
Address : 10_H
Reset Value : 00000000_H



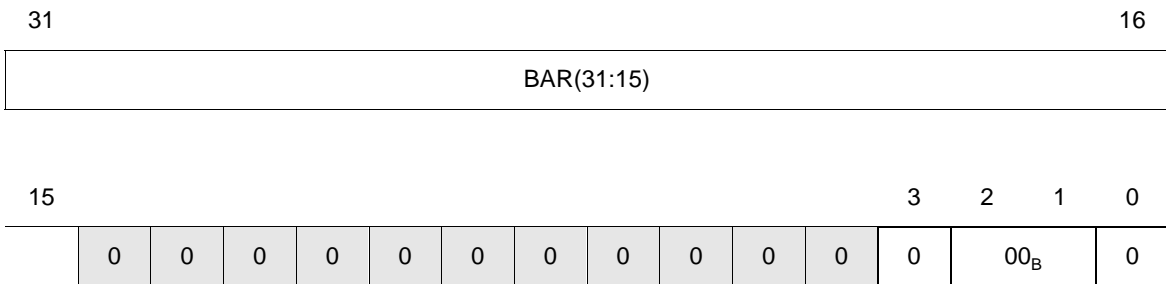
The first base address of the MUNICH256FM is marked as non-prefetchable and can be relocated anywhere in 32 bit address space of PCI memory. The MUNICH256FM supports memory accesses only.

BAR **Base Address**

The base address will be used for determining the address space of the MUNICH256FM and to do the mapping of the address space. Since the device allocates a total of 4 kByte address space BAR(31:12) are implemented as read/writable.

BAR2
Base Address 2

Access : read/write
Address : 14_H
Reset Value : 00000000_H



The second base address of the MUNICH256FM is marked as non-prefetchable and can be relocated anywhere in 32 bit address space of PCI memory. The MUNICH256FM supports memory accesses only. All accesses to memory regions defined by BAR2 will be mapped to the local bus.

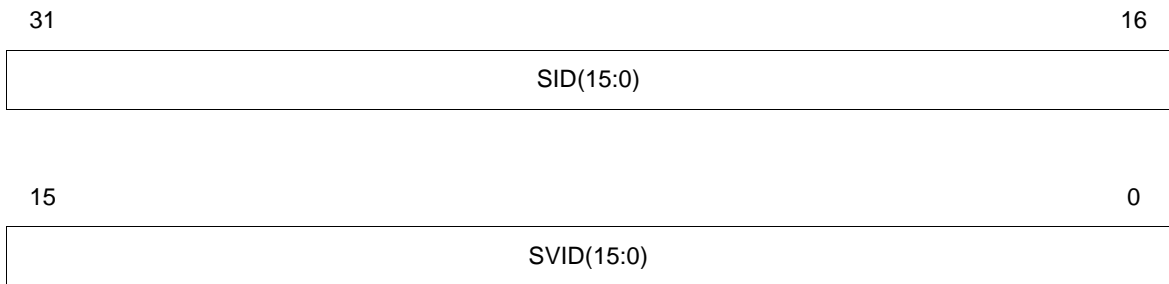
BAR **Base Address**

The base address will be used for determining the address space of the memory regions located on the local bus of the MUNICH256FM and to set the mapping of the address space. The MUNICH256FM can access a total of 24 kByte address space on the local bus as a bus master.

In those applications where the master functionality of MUNICH256FM is not needed the second base address register BAR2 may be disabled using bit MEM.BAR2 in the PCI user configuration space.

SID/SVID
Subsystem ID/Subsystem vendor ID

Access : read
Address : 2C_H
Reset Value : 00000000_H



SID Subsystem ID
The subsystem ID uniquely identifies the add-in board or subsystem where the system resides. The value of SID may be reconfigured after the reset phase of the system via the SPI interface.

SVID Subsystem Vendor ID
The subsystem vendor ID identifies the vendor of an add-in board or subsystem. The value may be reconfigured after the reset phase of the system via the SPI interface.

ML/MG/IP/IL

Maximum Latency/Minimum Grant/Interrupt Pin/Interrupt Line

Access : read/write
 Address : 3C_H
 Reset Value : 06020100_H



- ML** Maximum Latency

This value specifies how often the device needs to access the PCI bus in multiples of 1/4 us. The value is hardwired to 06_H.
- MG** Minimum Grant

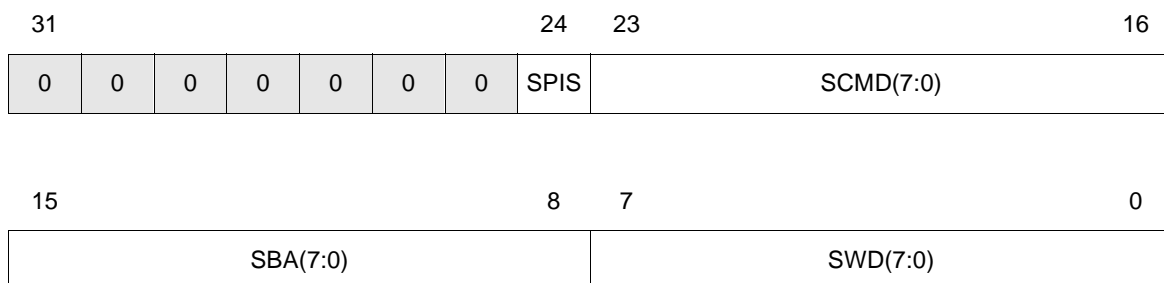
This value specifies how long of a burst period the device needs, assuming a clock rate of 33 MHz in multiples of 1/4 us. The value is hardwired to 02_H.
- IP** Interrupt Pin

The interrupt pin register tells which interrupt pin the device uses. Refer to section 6.2.4 and to section 2.2.6 of the PCI specification Rev. 2.1. The value is hardwired to 01_H.
- IL** Interrupt Line

The interrupt line register is used to communicate interrupt line routing information.

SPI
SPI Access Register

Access : read/write
Address : 40_H
Reset Value : 0000001F_H



SPIS SPI Start
To start the EEPROM transaction, which is defined in the SPI command, the byte address, and the data field, this bit must be set to '1' by a write transaction through the PCI interface. After the transaction is finished, the start bit is deasserted by the SPI interface controller. This signal must be polled by system software.

SCMD SPI Command
In this register, the SPI command for the next EEPROM transfer must be written before the transaction is started. The following SPI commands are supported:

- | | | |
|-----------------|-------|-----------------------------|
| 01 _H | WRSR | Write Status Register |
| 02 _H | WRITE | Write Data to Memory Array |
| 03 _H | READ | Read Data from Memory Array |
| 04 _H | WRDI | Reset Write Enable Latch |
| 05 _H | RDSR | Read Status Register |
| 06 _H | WREN | Set Write Enable Latch |

SBA SPI Byte Address
For read and write transaction to the connected EEPROM, the byte address must be written in this register before the transaction is started.

Register Description

SD

SPI Data

For the write status register transactions and the write data to memory array transactions, the data, that has to be written to the EEPROM, must be written to this register before the transaction is started. After a read status register transaction or read data from memory array transaction has finished (start bit is deasserted), the byte received from the EEPROM is available in this register.

LR
Long Request Register

Access : read/write
Address : 44_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|----|---|----|
| 31 | | | | | | | | | | | | | 16 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | | | | | | | | | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LR |

- LR Long Request
- 0 The PCI interface deasserts the $\overline{\text{REQ}}$ signal in parallel with the assertion of the $\overline{\text{FRAME}}$ signal.
 - 1 The $\overline{\text{REQ}}$ signal will be deasserted in parallel with the deassertion of $\overline{\text{FRAME}}$.

MEM
PCI Memory Command Register

Access : read/write
Address : 48_H
Reset Value : 000007E6_H

| | | | | | | | | | | | | | | | | | | | |
|----|----|---|---|---------|---|---|----------|----|---|---------|---|----|----|------|---|---|--|--|---|
| 31 | 30 | | | | | | | | | | | 17 | 16 | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | |
| | | | | | | | | | | | | | | BAR2 | 0 | | | | |
| | | | | 15 | | | | 11 | | | 8 | 7 | | | 4 | 3 | | | 0 |
| 0 | 0 | 0 | 0 | MW(3:0) | | | MRL(3:0) | | | MR(3:0) | | | | | | | | | |

BAR2 Enable Base Address Register 2
Setting this bit enables Base Address Register 2. Per default base address register two is disabled. If an EEPROM is connected to the SPI interface the value of this bit can be loaded via the EEPROM. Additionally this bit can set using standard PCI configuration write commands.

- 0 Base Address Register 2 is disabled.
- 1 Base Address Register 2 is enabled.

MW Memory Write Command
The value of this register contains the write command to be used during initiator transfers and is set to memory write after reset. The value of this register is configurable during setup of the bridge either by loading the value from EEPROM or by writing from PCI side.

MRL Memory Read Command (Long transfers)
The value of this register defines command to be used for read transfers which are equal or more than two DWORDs and is set to memory read line after reset. The value of this register is configurable during run time of the bridge either by loading the value from EEPROM or by writing from PCI side.

MR Memory Read Command
The value of this register defines command to be used for read transfers of single DWORDs. The value of this register is configurable during run

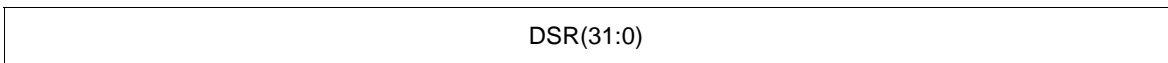
Register Description

time of the bridge either by loading the value from EEPROM or by reading or writing from PCI side.

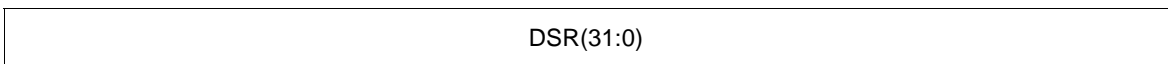
DEBUG
PCI Debug Support Register

Access : read
Address : 4C_H
Reset Value : 00000000_H

31 16



15 0



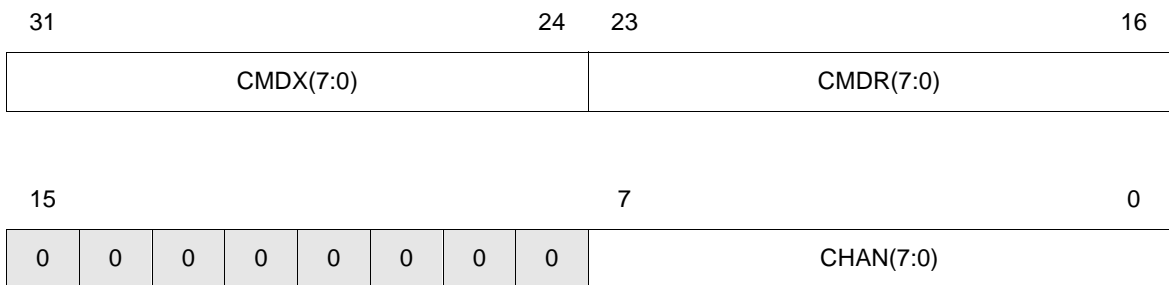
DSR Debug Support register

The value of this register contains the address of the next initiator transfer during normal operation. In case of disconnect, retry, master abort and target abort the register contains the address of the failed transaction.

8.2.2 PCI Slave Register

CSPEC_CMD Channel Specification Command Register

Access : read/write
Address : 000_H
Reset Value : 00000000_H



The channel specification registers are the access registers to the chip internal channel database. In order to program or reprogram a channel the channel information must be setup in the channel specification data registers before a channel command can be given. As soon as the channel command is issued the channel information is copied to the chip internal channel database and the device is reconfigured for the intended operation. Since reconfiguration time is dependent on the given command, certain commands generate acknowledge/fail command interrupt vectors to report status of configuration. During this time (command has been given and command interrupt) no further commands are allowed for the same channel. Please note that any command for one channel does not affect operation of any other channel.

For configuration of multiple channels the system software needs to program the channel data registers only once and then can issue channel commands for multiple channels without reprogramming the channel data registers.

Note: Debugging of channel information using the commands 'Receive Debug' or 'Transmit Debug' requires new programming of channel data registers for further operation.

For detailed description of register concept and command concept refer to chapter "Channel Programming / Reprogramming Concept" on page 162.

Register Description

| | |
|------|---|
| CMDX | <p>Command Transmit</p> <p>For detailed description of transmit commands and programming sequences refer to Chapter 6.2.</p> <p>01_H Transmit Init</p> <p>02_H Transmit Off</p> <p>04_H Transmit Abort/Branch</p> <p>08_H Transmit Hold Reset</p> <p>10_H Transmit Debug</p> <p>20_H Transmit Idle</p> <p>40_H Transmit Update</p> |
| CMDR | <p>Command Receive</p> <p>For detailed description of receive commands and programming sequences refer to Chapter 6.3.</p> <p>01_H Receive Init</p> <p>02_H Receive Off</p> <p>04_H Receive Abort/Branch</p> <p>08_H Receive Hold Reset</p> <p>10_H Receive Debug</p> |
| CHAN | <p>Channel select</p> <p>0..255 Selects the channel to be programmed or debugged.</p> |

Note: Transmit init for a channel must be programmed only after reset or after a transmit off command, i.e. two transmit init commands for the same channel are not allowed.

CSPEC_MODE_REC
Channel Specification Mode Receive Register

Access : read/write
Address : 004_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|----|------|-----|-----|------------|------|-----------|------------|------------|---|---|---|---|---|----------|----|
| 31 | | | 28 | 27 | | 24 | 23 | | | | | | | | 16 |
| 0 | 0 | 0 | DEL | ACCMX(3:0) | | | | RFLAG(7:0) | | | | | | | |
| | | | | | | | | | | | | | | | |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | | | | | | 1 | 0 |
| 0 | SFDE | TFF | INV | TMP | CRCX | CRC 32 | CRC DIS | 0 | 0 | 0 | 0 | 0 | 0 | PMD(1:0) | |

- DEL** DEL (Delete) Demap
This bit enables demapping of the control character DEL. This bit is valid in PPP modes only.
0 Disable demapping of control character DEL.
1 Enable demapping of control character DEL.
- ACCMX** Extended ACCM
In addition to the *Channel Specification Receive ACCM Map* the user can select four global user definable characters for character demapping in PPP modes. Setting one or more of the bits ACCM(3) through ACCM(0) enables the corresponding character which can be found in register REC_ACCMX.
0 Disable the selected character in REC_ACCMX for character demapping.
1 Enable the corresponding character in register REC_ACCMX for character demapping.
- RFLAG** Receive Flag
Used in transparent mode only. The RFLAG constitutes the flag that is filtered from the received bit stream if enabled via bit TFF.

Register Description

| | |
|------|--|
| SFDE | <p>Short/Small Frame Drop Enable</p> <p>This bit enables either the drop of short frames or the drop of small frames. This bit is valid in HLDC and PPP modes only.</p> <p>0 Short Frame Drop. Frames smaller than four bytes payload data (CRC32) or smaller than two bytes payload data (CRC16) are dropped. This function is not available if bit CRCX is enabled.</p> <p>1 Small Frame Drop. Frames (Payload and CRC) which are smaller or equal to CONF3.MINFL are dropped.</p> |
| TFF | <p>TMA Flag</p> <p>This bit enabled flag extraction in TMA mode and is available if non of the bits belonging to this channel is masked.</p> <p>0 No flag extraction</p> <p>1 Enable flag extraction. The flag specified in RFLAG will be extracted from the received data stream.</p> |
| INV | <p>Bit Inversion</p> <p>When bit inversion is enabled incoming channel data is inverted before processed by the protocol machine. E.g. incoming octet 81_H will be recognized as idle flag in HDLC mode.</p> <p>0 No Bit Inversion</p> <p>1 Bit Inversion</p> |
| TMP | <p>Transparent Mode Packing</p> <p>This bit enables the transparent mode packing and is valid in TMA mode only. This feature is applicable if at least one bit in any time slot is masked.</p> <p>0 Incoming masked bits are substituted with '1'. The non-used (masked) data bits are substituted by '1's.</p> <p>1 If subchanneling is used in transparent mode (i.e. less than 8 bits of a time slot are used), the non-used (masked) data bits are discarded.</p> |
| CRCX | <p>CRC Transfer</p> <p>This bit enables the capability to store the CRC checksum of incoming data packets in system memory together with the payload data.</p> <p>0 The CRC checksum from the incoming data packet will be removed from the packet and not transferred to the shared memory.</p> <p>1 The CRC checksum together with the payload data is transferred to the shared memory.</p> |

Register Description

| | |
|--------|---|
| CRC32 | CRC32 Select |
| | This bit selects the generator polynomial in the receiver. The checksum of incoming data packets will be compared against CRC16 or CRC32. CRC Select is valid in HDLC and PPP modes only. |
| | 0 Select CRC16 checksum. |
| | 1 Select CRC32 checksum. |
| CRCDIS | CRC Check Disable |
| | This bit disables CRC Check in HDLC and PPP protocol modes. |
| | 0 CRC check is enabled. |
| | 1 CRC check is disabled. |
| PMD | Protocol Machine Mode |
| | These bit fields select the protocol machine mode in receive direction. |
| | 00 _B Select HDLC operation. |
| | 01 _B Select Bit synchronous PPP. |
| | 10 _B Select Byte synchronous PPP. |
| | 11 _B Select Transparent Mode. |

CSPEC_REC_ACCM
Channel Specification Receive ACCM Map Register

Access : read/write
Address : 008_H
Reset Value : 00000000_H

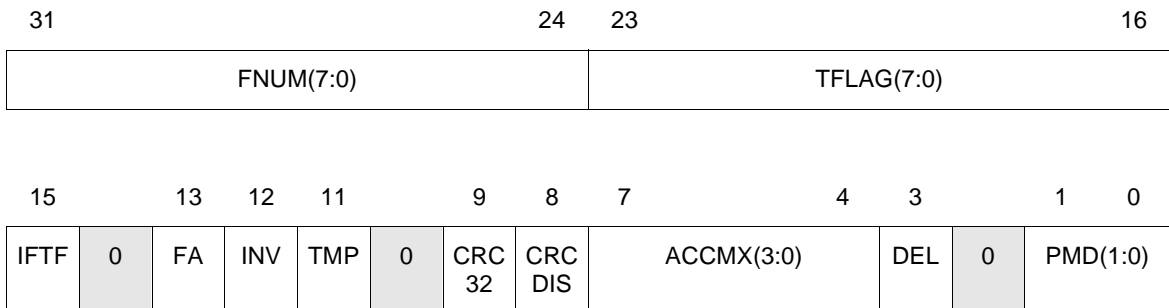
| | | | | | | | | | | | | | | | | |
|----|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 31 | | | | | | | | | | | | | | | | 16 |
| | 1F _H | 1E _H | 1D _H | 1C _H | 1B _H | 1A _H | 19 _H | 18 _H | 17 _H | 16 _H | 15 _H | 14 _H | 13 _H | 12 _H | 11 _H | 10 _H |
| | | | | | | | | | | | | | | | | 0 |
| 15 | 0F _H | 0E _H | 0D _H | 0C _H | 0B _H | 0A _H | 09 _H | 08 _H | 07 _H | 06 _H | 05 _H | 04 _H | 03 _H | 02 _H | 01 _H | 00 _H |

Any of the given characters can be selected for character demapping. If a bit is set the corresponding character is expected to be mapped by the control ESC character and is removed if received. These bits are valid in octet synchronous PPP modes only.

*Note: If this register needs to be reprogrammed, it must be done **before** accessing the register CSPEC_MODE_REC.*

CSPEC_MODE_XMIT
Channel Specification Mode Transmit Register

Access : read/write
Address : 014_H
Reset Value : 00000000_H



- FNUM** Flag number
FNUM denotes the number of flags send between two frames. The flag number can be updated during transmission with command 'Transmit Update'.
0 One flag is sent between two frames (shared flag).
1..255 FNUM+1 flags are sent between two frames.
- TFLAG** Transparent flag
Only valid if transparent mode is selected and if FA is enabled. TFLAG constitutes the flag that is inserted into the transmit bit stream.
- IFTF** Interframe Time Fill
This bit determines the interframe time fill in HDLC and PPP modes.
0 Interframe time fill is 7E_H.
1 Interframe time fill is FF_H.
- FA** Flag Adjustment
Only valid if transparent mode is selected.
0 The value FF_H is sent in sent in all TMA mode exception conditions.
1 The value specified in TFLAG is sent in all TMA mode exception conditions (e.g. idle). This bit can be set only when none of the bits belonging to this channels is masked.

Register Description

| | |
|--------|---|
| INV | <p>Bit Inversion</p> <p>If bit inversion is enabled outgoing channel data is inverted after processed by the protocol machine. E.g. a outgoing idle flag is transmitted as octet 81_H in HDLC mode.</p> <p>0 Disable bit inversion.</p> <p>1 Enable bit inversion.</p> |
| TMP | <p>Transparent Mode Pack</p> <p>This bit enables the transparent mode packing and is valid in TMA mode only. This feature is applicable if at least one bit in any time slot is masked.</p> <p>0 If subchanneling is used outgoing masked bits of data octet are discarded and substituted with '1'.</p> <p>1 If subchanneling is used outgoing masked bits are sent as '1'. The remaining bits of data are sent in the next time slot.</p> |
| CRC32 | <p>CRC 32 Select</p> <p>This bit selects the generator polynomial in the transmitter. The checksum of outgoing data packets will be generated according to CRC16 or CRC32. CRC32 Select is valid in HDLC and PPP modes only.</p> <p>0 Select CRC16 generation.</p> <p>1 Select CRC32 generation.</p> |
| CRCDIS | <p>CRC Disable</p> <p>This bit enables generation and transmission of a CRC checksum. CRC disable is valid in HDLC and PPP modes only.</p> <p>0 CRC generation and transmission is disabled.</p> <p>1 CRC generation and transmission is enabled.</p> |
| ACCMX | <p>Enable extended ACCM character</p> <p>The selected bits in bit field ACCMX denote the enabled characters in XMIT_ACCMX.</p> <p>In addition to the <i>Channel Specification Transmit ACCM Map</i> the user can select four global user definable characters for character mapping in PPP modes. Setting one or more of the bits ACCM(3) through ACCM(0) enables the corresponding character which can be found in register XMIT_ACCMX.</p> <p>0 Disable the selected character in XMIT_ACCMX for character mapping.</p> <p>1 Enable the corresponding character in register XMIT_ACCMX for character mapping.</p> |

Register Description

| | |
|-----|--|
| DEL | <p>DEL (Delete) Map Flag</p> <p>This bit enables mapping of the control character DEL. This bit is valid in PPP modes only.</p> <p>0 Disable mapping of DEL.</p> <p>1 Enable mapping of DEL.</p> |
| PMD | <p>Protocol Machine Mode</p> <p>This bit field selects the protocol machine mode in transmit direction.</p> <p>00_B Select HDLC operation.</p> <p>01_B Select Bit synchronous PPP.</p> <p>10_B Select Byte synchronous PPP.</p> <p>11_B Select Transparent Mode.</p> |

CSPEC_XMIT_ACCM
Channel Specification Transmit ACCM Map Register

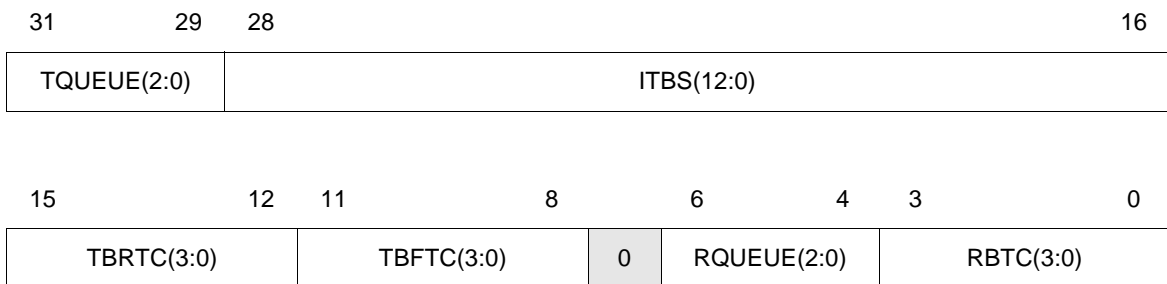
Access : read/write
Address : 018_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| 31 | | | | | | | | | | | | | | | 16 |
| 1F _H | 1E _H | 1D _H | 1C _H | 1B _H | 1A _H | 19 _H | 18 _H | 17 _H | 16 _H | 15 _H | 14 _H | 13 _H | 12 _H | 11 _H | 10 _H |
| | | | | | | | | | | | | | | | |
| 15 | | | | | | | | | | | | | | | 0 |
| 0F _H | 0E _H | 0D _H | 0C _H | 0B _H | 0A _H | 09 _H | 08 _H | 07 _H | 06 _H | 05 _H | 04 _H | 03 _H | 02 _H | 01 _H | 00 _H |

Any of the given characters can be selected for character mapping. If a bit is set the corresponding character will be mapped by the control ESC character. These bits are valid in octet synchronous PPP modes only.

CSPEC_BUFFER Channel Specification Buffer Configuration Register

Access : read/write
Address : 020_H
Reset Value : 00200000_H



- TQUEUE** Transmit Interrupt Vector Queue
This bit field determines the interrupt queue where channel interrupts transmit will be stored.
- ITBS** Individual transmit buffer size
Note: Please note that the internal architecture is 32 bit wide. Therefore each buffer location corresponds to four data octets.
The transmit buffer size configures the number of internal transmit buffer locations for a particular channel. Buffer locations will be allocated on command transmit init and released after command transmit off.
Note: The sum of transmit forward threshold and transmit refill threshold must be smaller than the internal buffer size.
- TBRTC** Transmit Buffer Refill Threshold Code
Note: Please note that the internal architecture is 32 bit wide. Therefore each buffer location corresponds to four data octets.
TBRTC is a coding for the transmit refill threshold. Please refer to **Table 8-7** for correspondence between code and threshold.
The internal transmit buffer has a programmable number of buffer locations per channel. When the number of free locations reach the transmit buffer refill threshold the internal transmit buffer requests new data from the data management unit.

Register Description

- TBFTC** Transmit Buffer Forward Threshold Code
Note: Please note that the internal architecture is 32 bit wide. Therefore each buffer location corresponds to four data octets.
- TBFTC is a coding for the transmit buffer forward threshold. Please refer to **Table 8-7** for correspondence between code and threshold.
- The transmit buffer forward threshold code determines the number of buffer locations which must be filled until protocol machine starts transmission. Nevertheless the transmit buffer forwards data packets to protocol machine as soon as a whole packet or the end of a packet is stored in the transmit buffer.
- RQUEUE** Receive Interrupt Queue.
 This bit field determines the interrupt queue number where channel interrupts receive will be stored.
- RBTC** Receive Buffer Threshold Code
Note: Please note that the internal architecture is 32 bit wide. Therefore each buffer location corresponds to four data octets.
- RBTC is a coding for the receive buffer threshold. Please refer to **Table 8-7** for correspondence between code and threshold.
- The receive buffer threshold determines the maximum packet size in DWORDs which will be stored in the internal receive buffer for a specific channel. When the packet size reaches the receive buffer threshold or a packet has been completely received, the packet will be forwarded to system memory.

Table 8-7 Threshold Codings

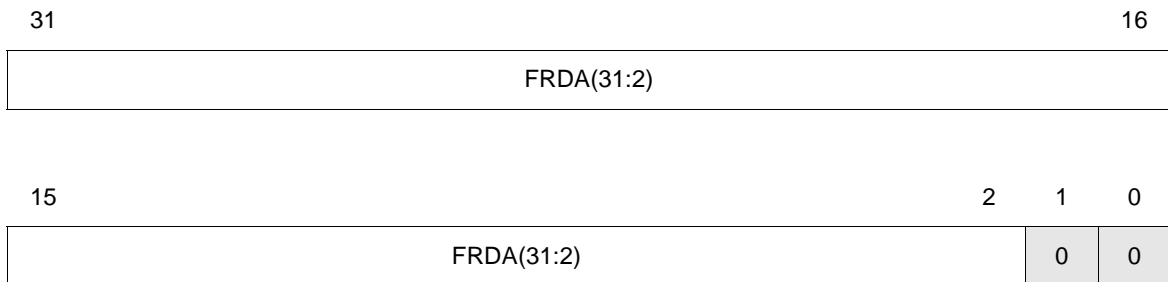
| Coding | Threshold in DWORDs | RBTC | TBFTC | TBRTC | TPBL |
|-------------------|---------------------|------|-------|-------|------|
| 0000 _B | 1 | x | x | x | x |
| 0001 _B | 4 | x | x | x | x |
| 0010 _B | 8 | x | x | x | x |
| 0011 _B | 12 | x | x | x | x |
| 0100 _B | 16 | x | x | x | x |
| 0101 _B | 24 | x | x | x | x |
| 0110 _B | 32 | x | x | x | x |
| 0111 _B | 40 | x | x | x | x |
| 1000 _B | 48 | x | x | x | x |

Register Description

| Coding | Threshold in DWORDs | RBTC | TBFTC | TBRTC | TPBL |
|-------------------|------------------------|-----------|-----------|-------|-----------|
| 1001 _B | 64 | x | x | x | x |
| 1010 _B | 96 | Not Valid | Not Valid | x | Not Valid |
| 1011 _B | 128 | | | x | |
| 1100 _B | 192 | | | x | |
| 1101 _B | 256 | | | x | |
| 1110 _B | 384 | | | x | |
| 1111 _B | 512 | | | x | |

CSPEC_FRDA
Channel Specification FRDA Register

Access : read/write
Address : 024_H
Reset Value : 00000000_H



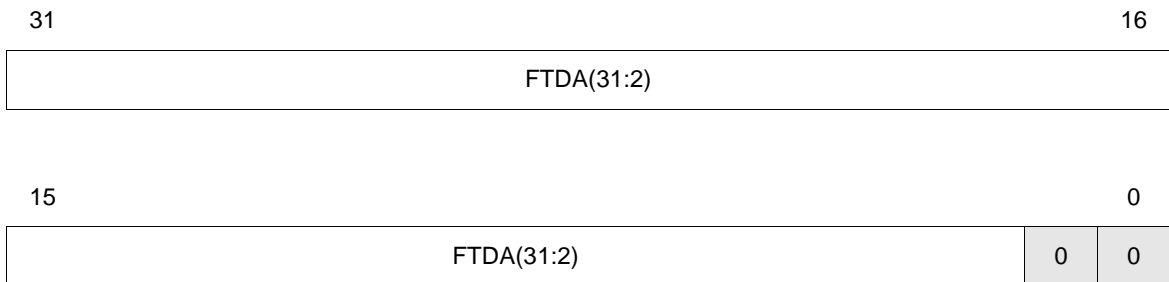
FRDA First Receive Descriptor Address

This 30-bit pointer contains the start address of the first receive descriptor. The receive descriptor is read entirely after the first request of the receive buffer and stored in the on-chip channel database. Therefore all information in the descriptor pointed to by FRDA must be valid when the data management unit branches to this descriptor.

The user can specify a new First Receive Descriptor Address using receive abort/branch command. In this case the First Receive Descriptor Address (FRDA) is used as a pointer to a new linked list. See details on commands in section "Channel Commands" on page 163.

CSPEC_FTDA
Channel Specification FTDA Register

Access : read/write
Address : 028_H
Reset Value : 00000000_H



FTDA First Transmit Descriptor Address

This 30-bit pointer contains the start address of the first transmit descriptor. The transmit descriptor is read entirely after the first request of the transmit buffer and stored in the on-chip channel database. Therefore all information in the descriptor pointed to by FTDA must be valid when the data management unit branches to this descriptor.

The user can specify a new First Transmit Descriptor Address using the 'Transmit Abort/Branch' command. In this case the first transmit descriptor address (FTDA) is used as a pointer to a new linked list. See details on commands in **Chapter 6.2**.

CSPEC_IMASK
Channel Specification Interrupt Vector Mask Register

Access : read/write
Address : 02C_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|----|-----|-----|------|-----|------|-----|------|------|-----|------|----|-----|----|---|-----|
| 31 | 30 | 28 | | | | 23 | | | | 22 | 16 | | | | |
| 0 | TAB | 0 | HTAB | 0 | 0 | 0 | 0 | UR | TFE | 0 | 0 | 0 | 0 | 0 | TCC |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 3 | | 2 | 0 | |
| 0 | RAB | RFE | HRAB | MFL | ROFD | CRC | ILEN | RFOP | SF | IFTC | 0 | SFD | SD | 0 | RCC |

For each channel or command related interrupt vector an interrupt vector generation mask is provided. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see **Chapter 4.13.1**.

The following definition applies:

- 1 The device will not generate the corresponding interrupt vector, i.e. the interrupt vector is masked.
- 0 An interrupt condition results in generation of the corresponding interrupt vector.

Channel Interrupt Vector Transmit

- TAB Mask 'Transmit Abort'
- HTAB Mask 'Hold Caused Transmit Abort'
- UR Mask 'Transmit Underrun'
- TFE Mask 'Transmit Frame End'

Command Interrupt Vector Transmit

- TTC Mask 'Transmit Command Complete'

Command Interrupt Vector Receive

| | |
|------|--|
| RAB | Mask 'Receive Abort' |
| RFE | Mask 'Receive Frame End' |
| HRAB | Mask 'Hold Caused Receive Abort' |
| MFL | Mask 'Maximum Frame Length Exceeded' |
| RFOD | Mask 'Receive Frame Overflow DMU' |
| CRC | Mask 'CRC Error' |
| ILEN | Mask 'Invalid Length' |
| RFOP | Mask 'Receive Frame Overflow' |
| SF | Mask 'Short Frame Detected' |
| IFTC | Mask 'Interframe Time-fill Flag' and 'Interframe Time-fill Idle' |
| SFD | Mask 'Short Frame Dropped' |
| SD | Mask 'Silent Discard' |
| RCC | Mask 'Receive Command Complete' |

CONF1 Configuration Register 1

Access : read/write
Address : 040_H
Reset Value : 820000F0_H

| | | | | | | | | | | | | | | | | | |
|-----------|---|---|---|---|---|------|------|------|------|------|-----------|-----|-----|---|--|--|----|
| 31 | | | | | | 25 | 24 | 23 | | | 21 | 20 | | | | | 16 |
| IIP | 0 | 0 | 0 | 0 | 0 | STOP | SRST | 0 | 0 | MFLE | MFL(12:0) | | | | | | |
| | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | | | |
| MFL(12:0) | | | | | | | MBIM | PBIM | RBIM | RFIM | SFL | RBM | LBE | 0 | | | |

- IIP** Initialization in Progress (Read Only)
After reset (hardware reset or software reset) the internal RAM's are self initialized by the MUNICH256FM. During this time (approx. 250 μ s) no other accesses to the device than reading register CONF1 or FCONF are allowed. This bit must be polled until it has been deasserted by the MUNICH256FM.
- 0 Self initialization has finished.
1 Self initialization in progress.
- STOP** Stop
After reset the MUNICH256FM can be switched to 'Fast Initialization' mode. During stop mode internal RAM's will not be accesses by internal state machines. This mode is for test purposes only and allows writing or reading the internal RAM's.
- 0 Device is in normal operation. This bit must be set to zero after chip initialization. See also "Mode Initialization" on page 169.
1 Device is in 'Fast Initialization Mode'. This function is used for test purposes only.
- SRST** Software Reset
This bit issues a software reset to the MUNICH256FM. During software reset all interfaces except PCI interface are forced into their idle state. After software reset is set the MUNICH256FM starts its self initialization

Register Description

and IIP will be asserted. When IIP is deasserted system software can reset SRST to '0' to start normal operation again.

- 0 Normal operation
- 1 Start software reset.

MFLE Maximum Frame Length Check Enable

- 0 Disable maximum frame length check.
- 1 Enable maximum frame length check.

MFL Maximum Frame Length

MFL defines the maximum length of incoming data packets. Packets exceeding the specified length are reported in the status field of the receive descriptor and if selected in an additional channel interrupt.

MBIM Mailbox Interrupt Vector Mask

This bit enables or disables mailbox system interrupt vectors generated by the mailbox.

- 0 Enable interrupt vector.
- 1 Disable interrupt vector.

PBIM PCI Bridge Interrupt Vector Mask

This bit enables or disables the 'PCI Access Error' interrupt vector generated by the PCI bridge.

- 0 Enable interrupt vector.
- 1 Disable interrupt vector.

RBIM Receive Buffer Interrupt Vector Mask

This bit enables or disables system interrupt vectors 'Receive Buffer Queue Early Warning' and 'Receive Buffer Action Queue Early Warning' which are generated by the receive buffer. RBIM is valid only if bit RBM is set.

- 0 Enable interrupt vector.
- 1 Disable interrupt vector.

RFIM Receive Buffer Failed Interrupt Vector Mask

This bit enables or disables the 'Receive Buffer Access Failed' interrupt vector.

- 0 Enable interrupt vector.
- 1 Disable interrupt vector.

Register Description

| | |
|-----|--|
| SFL | <p>Short Frame Length</p> <p>This bit is a global parameter which defines the length of short frames for all channels.</p> <p>0 Short frame is defined as a frame containing less than 4 bytes (CRC16) or less than 6 bytes (CRC32).</p> <p>1 Short frame is defined as a frame containing less than 2 bytes (CRC16) or less than 4 bytes (CRC32).</p> |
| RBM | <p>Receive Buffer Monitor</p> <p>This bit is provided to switch between two monitoring functions of the receive buffer. Receive buffer monitor functions are available in register RBTH and RBMON.</p> <p>0 The minimum free pool count is captured in register RBTH.</p> <p>1 An interrupt is generated, if the free pool counter falls below the value programmed in register RBTH.</p> |
| LBE | <p>Little/Big Endian Byte Swap</p> <p>This bit enables the little or big endian mode, which affects the data structures pointed to by data pointer of receive or transmit descriptor in system memory. Registers, interrupt vectors or descriptors are not affected by little/big endian byte swap.</p> <p>0 Switch data section to little endian mode.</p> <p>1 Switch data section to big endian mode.</p> |

CONF2 Configuration Register 2

Access : read/write
Address : 044_H
Reset Value : 00000000_H

| | | | | | | | | | | |
|-----|-----------|----|-----------|------------|-----------|----|-----|-------|----------|----|
| 31 | 30 | 28 | 27 | 26 | 24 | 23 | 22 | 21 | 20 | 16 |
| 0 | SYSQ(2:0) | | 0 | PORTQ(2:0) | | | TBE | RSPEN | SPA(4:0) | |
| 15 | 13 | 12 | 8 | 7 | 0 | | | | | |
| RCL | 0 | 0 | LPID(4:0) | | LCID(7:0) | | | | | |

- SYSQ** System Interrupt Queue
SYSQ sets up the interrupt queue where system interrupt vectors will be written to. One system interrupt queue can be selected for system interrupts.
- PORTQ(2:0)** Port Interrupt Vector Queue
PORTQ sets up the interrupt queue where port interrupt vectors will be written to. One interrupt queue can be selected for port interrupts.
- TBE** Test Breakout Enable
This bit enables the test breakout function. The incoming signals of the port selected via LPID are switched to the test ports and the incoming signals on the test port replace the output signals of the selected port. Setting TBE enables the selected port (tri-state no longer active) and has priority over functions selected in register PMR and priority over bit RSPEN. The port may be disabled using register REN and TEN to disable internal processing while test function is active.
- 0 Disable test function.
1 Enable test function.
- RSPEN** Receive Synchronization Pulse Enable
0 The selected transmit clock of port zero is visible on pin TCLKO. This function is available when port zero is operated in unchannelized mode.

Register Description

| | | |
|------|---|--|
| | 1 | The internally generated synchronization pulse of input port CONF2.SPA is switched to pin RSPO for test purposes. |
| SPA | | Synchronization Pulse Access This bit field selects one framer 0..27 whose synchronization pulse can be externally monitored. Only valid if RSPEN is set. |
| RCL | | Remote Channel Loop The remote channel loop switches incoming data of one channel to the outgoing bit stream of the same channel. The bit rate of the receiver and the transmitter must be the same. The channel to be looped can be selected using bit field LCID. One channel at a time can be looped. 0 Disable remote channel loop. 1 Enable remote channel loop. |
| LPID | | Port Identifier This bit field selects the port which shall be switched to the test port. See also bit CONF1.TBE. |
| LCID | | Loop Channel Identifier This bit field selects the channel which shall be looped through the internal loop buffer. |

CONF3 Configuration Register 3

Access : read/write
Address : 048_H
Reset Value : 00090000_H

| | | | | | | | | | | | | | | |
|----|----|------------|---|---|---|---|---|---|---|---|---|-----------|----|---|
| 31 | | | | | | | | | | | | 19 | 16 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TPBL(3:0) | | |
| | | | | | | | | | | | | 8 | 0 | |
| 15 | 13 | MINFL(5:0) | | | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

TPBL Transmit Packet Burst Length
This bit field is a coding for the maximum burst length on PCI bus, when data management unit fetches transmit packets. Please refer to **Table 8-7 "Threshold Codings" on page 208** for correspondence between code and maximum burst length.

MINFL Minimum Frame Length
Only valid for those channel which have bit CSPEC_MODE_REC.SFDE set. MINFL sets the minimum frame length in bytes (payload bytes and CRC bytes) for frames which will be forwarded to system memory. If enabled the receive buffer will drop frames which are smaller or equal to the programmed value MINFL to avoid wasting of PCI bandwidth in case of error conditions. The small frame check is disabled, if MINFL is set to zero.

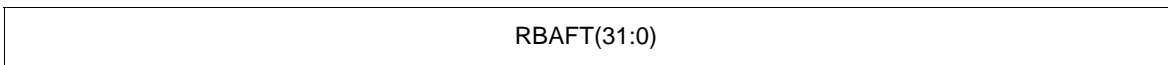
Note: Since the receive packets will be dropped inside the receive buffer, the receive packet threshold CSPEC_BUFFER.RTC has to be greater than MINFL/4 in order to work properly.

RBAFT
Receive Buffer Access Failed Interrupt Threshold Register

Access : read/write
Address : 04C_H
Reset Value : 00000000_H

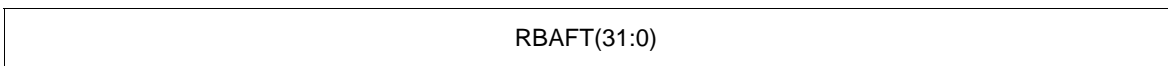
31

16



15

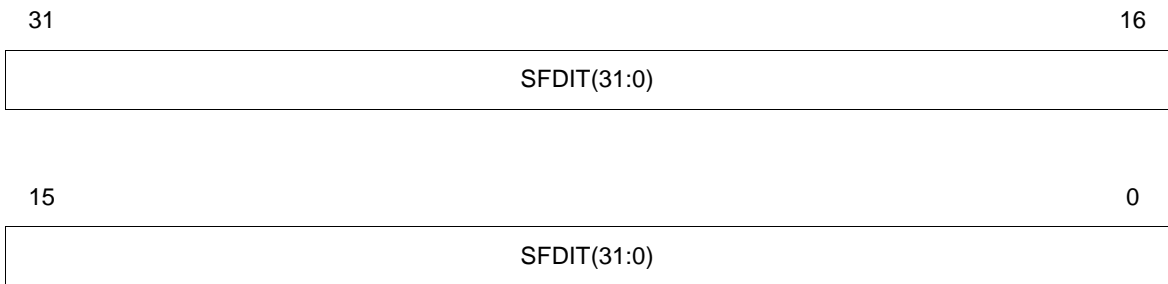
0



RBAFT **Receive Buffer Access Failed Interrupt Threshold**
This register sets the threshold for the 'Receive Buffer Access Failed' interrupt vector.

SFDT
Small Frame Dropped Interrupt Threshold Register

Access : read/write
Address : 050_H
Reset Value : 00000000_H

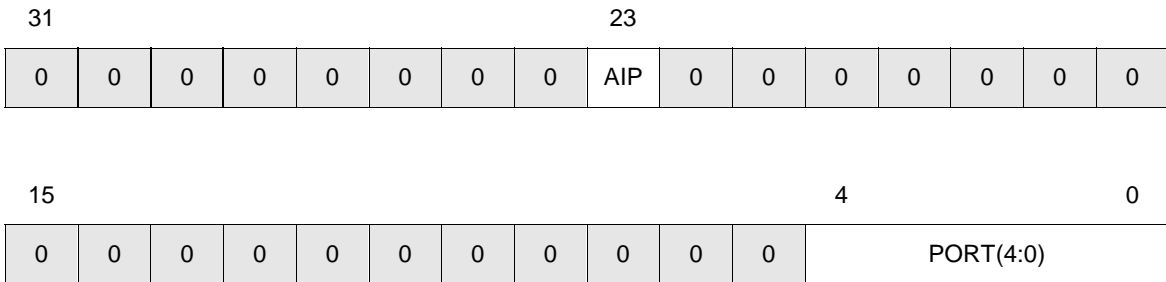


SFDIT Small Frame Dropped Interrupt Vector Threshold

The programmed threshold defines the threshold for the 'Small Frame Dropped' interrupt vector. As soon as the internal number of dropped, small frames reaches the programmed value a channel interrupt vector with bit SFD set will be generated. The actual value of dropped frames can be read using register SFDC. The value is applied to all 256 channels.

PMIAR
Port Mode Indirect Access Register

Access : read/write
Address : 060_H
Reset Value : 00000000_H



Note: This register is an indirect access register which must be programmed before accessing the register PMR.

- AIP** Auto Increment Port
 This bit enables the auto increment function of bit field PORT. Each read/write access to register PMR increments PORT. This allows to program multiple, consecutive ports without accessing PMIAR again.
 0 Disable auto increment function.
 1 Enable auto increment function.
- PORT** Port Select
 This bit field selects the port number, which can be accessed via register PMR.
 0..27 Port Number

PMR
Port Mode Register

Access : read/write
Address : 064_H
Reset Value : 0104C000_H

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------|-----|---|-----|----|---|-------|------------------|-----|-----|-----|----|---|------------------|---|----|----|----|----|----|----|----|---|---|---|---|---|--|--|--|---|
| 31 | | | | 28 | | | | 24 | | | 22 | | | | 18 | | | 16 | | | | | | | | | | | | |
| PCM(3:0) | | | | 0 | 0 | 0 | 000 _B | | | 0 | 0 | 0 | 000 _B | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | | | | 0 |
| RIM | TIM | 0 | TXR | 0 | 0 | CTFSD | LT | RLL | RPL | LPL | 0 | 0 | 0 | 0 | 0 | | | | | | | | | | | | | | | |

Note: Effected port is selected via register PMIAR. All settings in this register affect the selected port only.

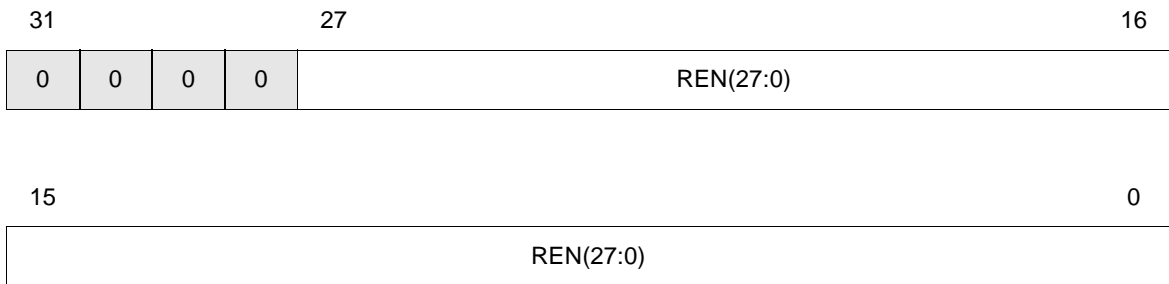
- PCM** Select Port Mode
This bit field selects the port mode.
0000_B T1 mode (1.544 MHz)
1000_B E1 mode (2.048 MHz)
1111_B Unchannelized mode
- RIM** Receive Synchronization Error Interrupt Vector Mask
This bit disables generation of the port interrupt vector receive. See "Port Interrupts" on page 127 for description of interrupt vectors.
0 Enable
1 Disable
- TIM** Transmit Synchronization Error Interrupt Vector Mask
This bit disables generation of the port interrupt vector transmit. See "Port Interrupts" on page 127 for description of interrupt vectors.
0 Enable
1 Disable

Register Description

| | |
|-------|---|
| TXR | <p>Transmit Data Rising</p> <p>This bit defines the edge the common transmit frame synchronization pulse CTFS is sampled on with respect to the common transmit clock CTCLK.</p> <p>0 CTFS is sampled on the rising edge of CTCLK.</p> <p>1 CTFS is sampled on the falling edge of CTCLK.</p> |
| CTFSD | <p>Common transmit frame synchronization disable</p> <p>0 Bit 0 of transmit data is synchronized to CTFS.</p> <p>1 Synchronization of data to CTFS is disabled.</p> |
| LT | <p>Looped Timing</p> <p>This bit selects the transmit clock in MUNICH256FM. Per default the transmit clock of the selected tributary is the common transmit clock. If set to '1' the corresponding tributary is switched into looped timed mode.</p> <p>0 Select normal operation mode.</p> <p>1 Select looped timing mode.</p> |
| RLL | <p>Remote Line Loop</p> <p>This bit enables the remote line loop of the selected port.</p> <p>0 Disable remote line loop.</p> <p>1 Enable remote line loop.</p> |
| RPL | <p>Remote Payload Loop</p> <p>This bit enables the remote payload loop of the selected port.</p> <p>0 Disable remote payload loop.</p> <p>1 Enable remote payload loop.</p> |
| LPL | <p>Local Port Loop</p> <p>This bit enables the local port loop on the selected port. When local loops are closed, the corresponding transmit clock and the synchronization pulse is switched to the receive port.</p> <p>0 Disable local port loop.</p> <p>1 Enable local port loop.</p> |

REN
Receive Enable Register

Access : read/write
Address : 068_H
Reset Value : 00000000_H



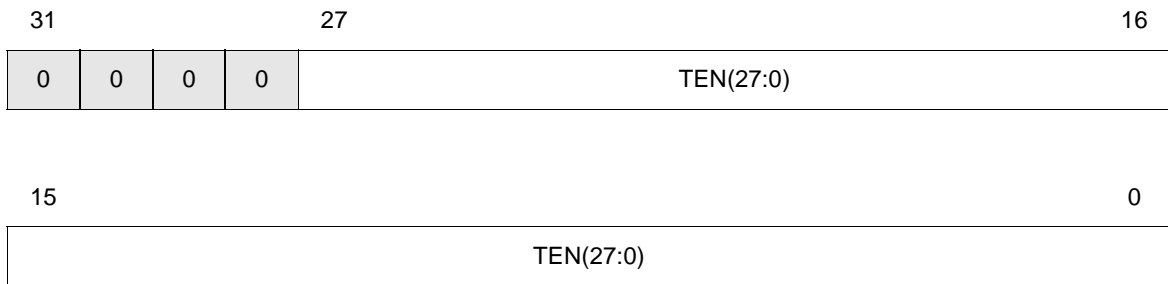
REN Receive Enable

Setting a bit in this bit field enables the receive function of the selected port. After reset all ports are disabled and thus all incoming receive data is discarded. While a port is disabled communication between port handler, time slot assigner and synchronization function is disabled. A port should be enabled if it is correctly configured using registers PMIAR and PMR.

0 Disable receive port.
1 Enable receive port.

TEN
Transmit Enable Register

Access : read/write
Address : 06C_H
Reset Value : 00000000_H



TEN **Transmit Enable**

This bit field enables the transmit function of the selected port. After reset all transmit ports are disabled and thus all TD lines are set to tri-state. While a port is reset the communication between port handler, time slot assigner and synchronization function is disabled. After the port mode has been selected using register PMIAR and PMR a transmit port can be enabled.

TSAIA
Time slot Assignment Indirect Access Register

Access : read/write
Address : 070_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | | |
|-----|---|---|---|-----------|----|---|---|---|-----|---|------------|---|---|---|---|----|
| 31 | | | | | | | | | | | 23 | | | | | 16 |
| DIR | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AIT | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | | | | | | | | | | | | | | | |
| 15 | | | | | 12 | | | | | 8 | | | 4 | | | 0 |
| | | | | PORT(4:0) | | | | | 0 | | TSNUM(4:0) | | | | | |

DIR Direction
 This bit select the direction for which programming is valid.
 0 Program time slots in receive direction.
 1 Program time slots in transmit direction.

AIT Auto Increment Time slot
 This bit enables the auto increment function of bit field TSNUM. Each read/write access to register TSAD increments TSNUM. This allows to program multiple, consecutive time slots without accessing TSAIA again.
 0 Disable auto increment function.
 1 Enable auto increment function.

PORT Port Select
 This bit field selects the port number, which can be accessed via register TSAIA.
 0..27 Port number

Register Description

TSNUM Time Slot Number

This bit field selects the time slots, which can be accessed via register TSAIA.

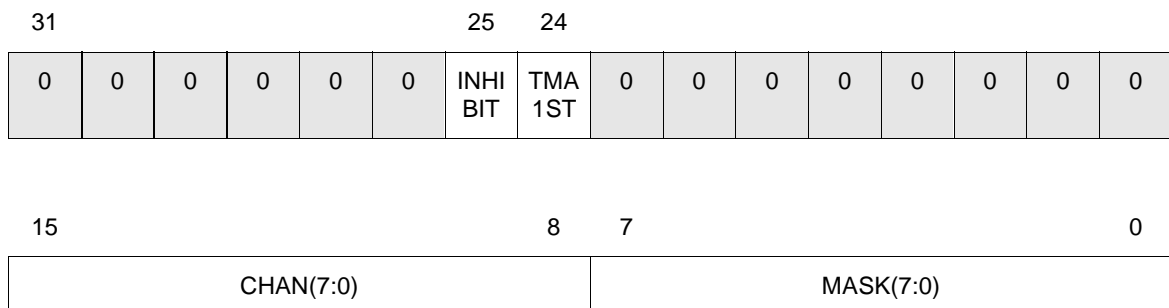
Valid time slot numbers are:

0..23 T1, Unchannelized

0..31 E1

TSAD
Time slot Assignment Data Register

Access : read/write
Address : 074_H
Reset Value : 02000000_H



Note: The time slot assignment data register assigns a channel and a mask to a specific port/time slot combination. The related port/time slot must be chosen by accessing TSAIA.

The time slot assignment has to be done before a specific channel is configured for operation. After operation the port/time slot assignment of a particular channel has to be set to inhibit.

INHIBIT Inhibit Time slot
This bit disabled processing of the selected port/time slot.
0 The time slot is enabled.
1 The time slot is disabled. In receive direction incoming octets are discarded. In transmit direction the octet of this time slot and port is set to FF_H.

TMA1ST TMA First
This bit marks the first time slot belonging to a TMA superchannel for TMA synchronization. Receiver starts processing data on the marked time slot. In transmit direction data transmission is started on the marked time slot. If TMA channel uses only one time slot this bit must be set.

CHAN Channel Number
This bit field selects the channel number which will be associated to the port and time slot which is selected in register TSAIA.

Register Description

MASK

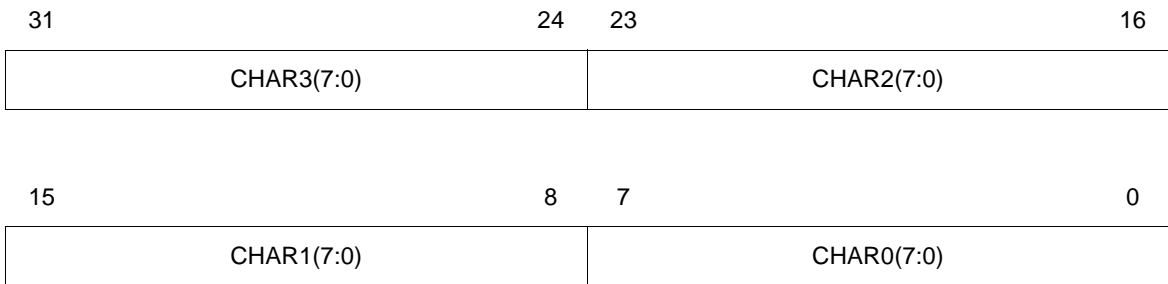
Mask Bits

Setting a bit in this bit field selects the corresponding bit in a time slot which is enabled for operation.

- 0 In receive direction the corresponding bit is discarded. In transmit direction the bit is sent as '1'.
- 1 In receive direction the corresponding bit is forwarded to the protocol machine (via time slot assigner). In transmit direction data on the serial line is generated by the protocol machine.

REC_ACCMX
Receive Extended ACCM Map Register

Access : read/write
Address : 080_H
Reset Value : 00000000_H

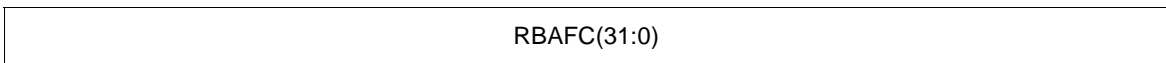


This register is only used by channels operated in octet synchronous PPP mode. A character written to this register is mapped with a control escape sequence, if the corresponding enable flag is set in the corresponding bit CSPEC_MODE_REC.ACCMX(3:0).

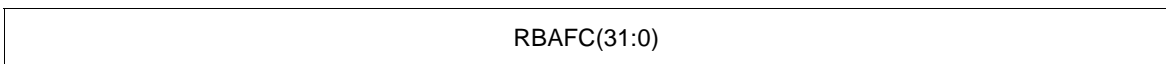
RBAFC
Receive Buffer Access Failed Counter Register

Access : read
Address : 084_H
Reset Value : 00000000_H

31 16



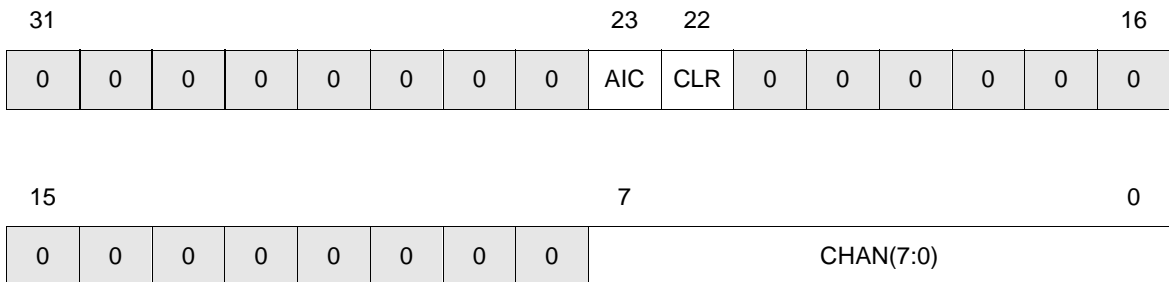
15 0



RBAFC **Receive Buffer Access Failed Counter**
The read value of this register defines the number of packets which have been discarded due to inaccessibility of the internal receive buffer. A read access resets the counter to zero.

SFDIA
Small Frame Dropped Indirect Access Register

Access : read/write
Address : 088_H
Reset Value : 00000000_H



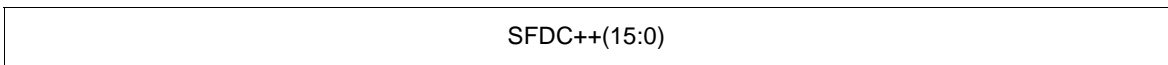
- AIC** Auto Increment Channel
This bit enables the auto increment function of bit field CHAN. Each read/write access to register SFD increments CHAN by two. This allows to read the status of multiple channels without accessing SFDIA again.
0 Disable auto increment function.
1 Enable auto increment function.
- CLR** Clear
This bit enables the counter mode on reads to register SFDC.
0 Read of register SFDC does not affect the small frame dropped counter.
1 After reading register SFDC the value of the small frame dropped counter will be reset to zero.
- CHAN** Channel Number
This bit field selects the channel, whose status can be read in register SFDC.
0..255 Channel number

SFDC
Small Frame Dropped Counter Register

Access : read
Address : 08C_H
Reset Value : 00000000_H

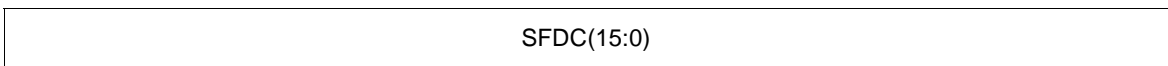
31

16



15

0

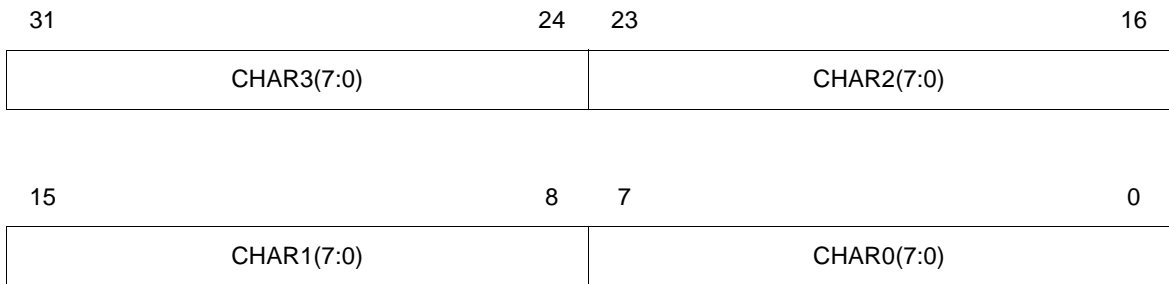


These both bit fields show the current value of the small frame dropped counter of the channel N and N+1 selected via SFDIA.CHAN. Dependent on bit field SFDIA.CLR the counter will be cleared after they are read.

- SFDC++ Small Frame Dropped Counter for Channel N+1
The number of dropped, small frames of channel SFDIA.CHAN+1.
- SFDC Small Frame Dropped Counter
The number of dropped, small frames of channel SFDIA.CHAN.

XMIT_ACCMX
Transmit Extended ACCM Map

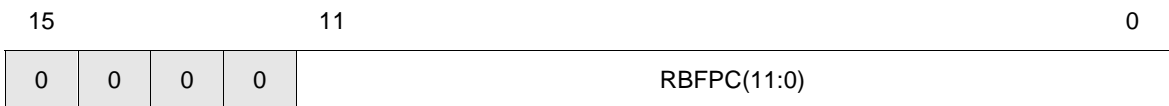
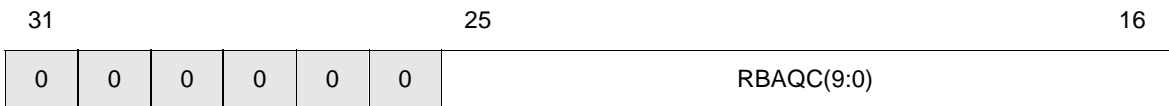
Access : read/write
Address : 090_H
Reset Value : 00000000_H



This register is only used by a channel in octet synchronous PPP mode. A character written to this register will be mapped with a Control Escape sequence, if the corresponding enable flag is set in the CSPEC_MODE_XMIT register (ACCMX(3:0)).

RBMON
Receive Buffer Monitor Indirect Access Register

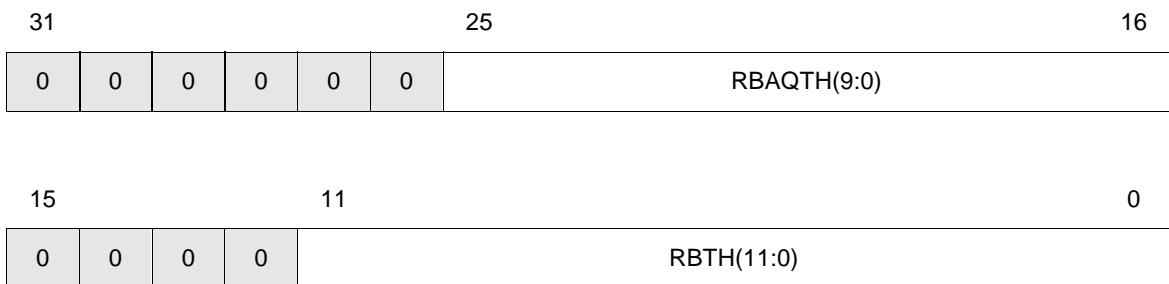
Access : read
Address : 0B0_H
Reset Value : 02000BFF_H



- RBAQC** Receive Buffer Action Queue Free Count
The value of this register determines the actual number of free actions inside the receive buffer.
- RBFPC** Receive Buffer Free Pool Count
The value of this register determines the actual number of free buffer locations inside the receive buffer. After reset a total number of 3072 receive buffer locations, which equals 12kB receive buffer, is available.

RBTH Receive Buffer Threshold Register

Access : read/write
Address : 0B4_H
Reset Value : 02000001_H



- RBAQTH** Receive Buffer Action Queue Free Pool Threshold
Function of RBAQTH is dependent on bit CONF1.RBM.
CONF1.RBM = '0':
The minimum value of RBMON.RBAQC, which occurred since the last reset or the last read of this register, is captured in here.
CONF1.RBM = '1':
A 'Receive Buffer Action Queue Early Warning' interrupt will be generated, if the receive buffer action queue free pool drops below the value programmed in bit field RBAQTH. The value to be programmed must be in the range of 000_H to 1FF_H.
- RBTH** Receive Buffer Free Pool Threshold
Function of RBTH is dependent on CONF1.RBM.
CONF1.RBM = '0':
The minimum value of RBMON.RBFP, which occurred since the last reset or the last read of this register, is captured in here.
CONF1.RBM = '1':
A 'Receive Buffer Queue Early Warning' interrupt vector will be generated, if the receive buffer free pool drops below the value programmed in bit field RBTH.

IQIA Interrupt Queue Indirect Access Register

Access : read/write
Address : 0E0_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|--------|------|------|-------|
| 31 | | | | | | | | | | | | | | | | 19 | 18 | 17 | 16 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DBG | SIQM | SIQL | SIQBA |
| | | | | | | | | | | | | | | | | 3 | | | 0 |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | Q(3:0) | | | |

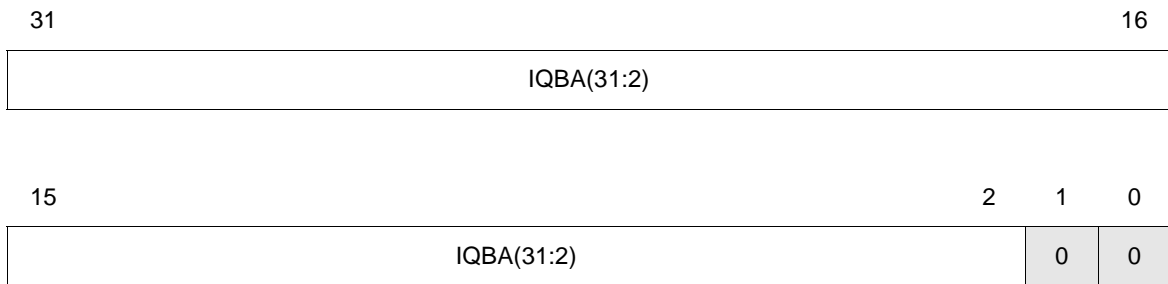
- DBG** Debug
- This bit selects the debug mode of the interrupt controller. When DEBUG is set, the actual values of interrupt queue base address, interrupt queue length and high priority interrupt queue mask of queue Q are copied to register IQBA, IQL and IQMASK. The value can be read with a following access to these registers.
- Note: Setting DEBUG is only allowed, if neither SIQBA, SIQL and SIQM are set.*
- 0 No operation
 - 1 Enable debug mode.
- SIQM** Set High Priority Interrupt Queue Mask
- This bit field enables setup of the high priority interrupt queue mask of queue Q. The value to be programmed has to be configured via register IQMASK prior to a write access to this bit.
- 0 No operation
 - 1 Set high priority mask.

Register Description

| | |
|-------|---|
| SIQL | <p>Set Interrupt Queue Length</p> <p>This bit field enables setup of the interrupt queue length of queue Q. The value to be programmed has to be configured via register IQL prior to a write access to this bit.</p> <p>0 No operation</p> <p>1 Set interrupt queue length.</p> |
| SIQBA | <p>Set Interrupt Queue Base address</p> <p>This bit field enables setup of the interrupt queue base address of queue Q. The value to be programmed has to be configured via register IQBA prior to a write access to this bit.</p> <p>0 No operation</p> <p>1 Update interrupt queue base address with value programmed in register IQBA.</p> |
| Q | <p>Interrupt Queue Number</p> <p>This bit field determines the interrupt queue number for which programming is valid. The first eight (0..7) interrupt queues are used for channel, port and system interrupt vectors, while the last interrupt queue (8) is used for command interrupt vectors. Interrupt queue number seven is per default the high priority interrupt queue.</p> <p>System software may setup the interrupt queue high priority mask, the interrupt queue length and the interrupt queue base address simultaneously by setting SIQL, SIQBA and SIQM.</p> <p>The command interrupt queue has a fixed length of two times 256 DWORDs, that is one DWORD for each interrupt vector.</p> <p>It is possible to setup the interrupt queue high priority mask, the interrupt queue length and the interrupt queue base address concurrently by setting SIQBA, SIQL and SIQM to '1'.</p> <p><i>Note: Programming of interrupt queue length or interrupt queue high priority mask is not valid for the command interrupt queue (interrupt queue 8).</i></p> <p><i>Note: Programming of interrupt queue high priority mask is not valid for the high priority interrupt queue (interrupt queue 7).</i></p> <p>0..8 Interrupt Queue</p> |

IQBA
Interrupt Queue Base Address Register

Access : read/write
Address : 0E4_H
Reset Value : 00000000_H



IQBA Interrupt Queue Base Address

The interrupt queue base address register assigns a base address to the eight channel interrupt queues and the command interrupt queue. To set a new base address for a specific queue, system software must first program IQBA. Afterwards the value is released by selecting the associated queue via bit field IQIA.Q and setting of bit IQIA.SIQBA. The interrupt queue base address has to be DWORD aligned. Whenever the base address of a particular interrupt queue is modified, the next interrupt vector written to that queue is stored in the first location of the queue.

IQL Interrupt Queue Length Register

Access : read/write
Address : 0E8_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|----------|---|---|---|---|----|---|---|
| 31 | | | | | | | | | | | | | 16 | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | | | | | | | | 7 | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IQL(7:0) | | | | | | | |

IQL Interrupt Queue Length

This bit field assigns a interrupt queue length to the eight channel interrupt queues. To set the interrupt queue length of a specific queue, system software must first program IQL. Afterwards the value is released by selecting the associated queue via bit field IQIA.Q and setting of bit IQIA.SIQL. IQL specifies the interrupt queue length L (number of DWORDs) in the shared memory with $L=(IQL+1)*16$ (maximum of 4092 DWORDs).

Note: IQL = 255 equals a queue length of 1 DWORD.

Whenever the length of a particular interrupt queue is modified, the next interrupt vector written to that queue is stored in the first location of the queue.

IQMASK Interrupt Queue High Priority Mask

Access : read/write
Address : 0EC_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|-----|-----|-----|------|-----|------|-----|------|------|-----|------|---|-----|----|---|----|
| 31 | 30 | | 28 | | | | | 23 | 22 | | | | | | 16 |
| THI | TAB | 0 | HTAB | 0 | 0 | 0 | 0 | UR | TFE | 0 | 0 | 0 | 0 | 0 | 0 |
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | | 3 | 2 | | 0 |
| RHI | RAB | RFE | HRAB | MFL | ROFD | CRC | ILEN | RFOP | SF | IFTC | 0 | SFD | SD | 0 | 0 |

In normal operation each channel interrupt vector is written to the interrupt queue associated with a specific channel, that is interrupt queue 0 to 7. The interrupt queue mask provides the functionality to forward selected channel interrupts to the high priority interrupt queue, which is hardwired as queue 7. Therefore a mask can be set for each of the interrupt queues, which specifies the channel interrupt vector to be forwarded to the high priority interrupt queue. To set the IQMASK for interrupt queues 0 to 6, system software must first program IQMASK. Afterwards the mask is released by selecting the affected interrupt queue via bit field IQIA.Q and setting of bit SIQM.

Those interrupt vectors which have an interrupt bit set, that is also masked in this high priority mask are forwarded to the high priority interrupt queue instead of the regular interrupt queue associated with a specific channel.

If a channel interrupt vector has at least one interrupt bit set, that is also masked in the high priority mask, the interrupt vector will be forwarded to the high priority interrupt queue.

In case that a channel interrupt vector has at least one bit set, that is not masked in the high priority mask, the interrupt vector is queued into the regular interrupt queue associated with the corresponding channel.

GISTA/GIACK Interrupt Status/Interrupt Acknowledge Register

Access : read/write
Address : 0F0_H
Reset Value : 00000000_H

| | | | | | | | | | | | | | | | |
|-------|---|---|---|---|---|---|----|----|----|----|----|----|----|-----|----|
| 31 | | | | | | | | | | | | | 17 | 16 | |
| INTOF | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LBI | IF |
| | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | Q8 | Q7 | Q6 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 |

Depending on the corresponding bits in register GMASK, an interrupt indication in this register will be flagged at pin INTA. If an interrupt bit is masked (set to '1') in register GMASK, system software has to poll this register in order to get status information of the disabled interrupt bit.

- INTOF** Interrupt Overflow
- This bit indicates that interrupt information has been lost due to overload conditions of the internal interrupt controller. This interrupt indicates a severe system problem. If this bit is set and INTOF is not masked in register GMASK, the interrupt pin INTA will be asserted. INTOF is cleared, when an '1' is written to this bit.
- 0 No interrupt overflow.
- 1 Interrupt overflow. The interrupt will be cleared by writing a '1' to the corresponding bit.
- LBI** Local Bus Interrupt
- The MUNICH256FM supports bridging of interrupts from the local bus to the PCI bus. In this application the pin LINT is used as an input and as

Register Description

soon as $\overline{\text{LINT}}$ changes from an inactive to an active state the interrupt pin $\overline{\text{INTA}}$ will be asserted.

Note: This bit does not clear by writing a '1'. This bit is set as long as the interrupt pin $\overline{\text{LINT}}$ is asserted.

0 $\overline{\text{LINT}}$ not asserted.

1 $\overline{\text{LINT}}$ asserted.

IF Interrupt FIFO

This bit indicates that there is an interrupt vector stored in the internal interrupt FIFO. The IF interrupt is available if the interrupt pin $\overline{\text{LINT}}$ is switched to input mode ($\text{INTCTRL.ID} = '1'$) and when the interrupt mask GMASK.IF is set to '0'.

Note: This bit does not clear by writing a '1'. This bit is set as long as an interrupt vector is stored in the interrupt FIFO.

0 No Interrupt vector in interrupt FIFO.

1 Interrupt vector stored in internal interrupt FIFO.

Q8..Q0 Interrupt Queue 8..0

On reads each bit flags one or more interrupt vectors that have been written to the corresponding interrupt queue. If one of the bits is set and the same bit is not masked in register GMASK , the interrupt pin $\overline{\text{INTA}}$ will be asserted. A bit is cleared, when an '1' is written to the specific bit.

0 No interrupt vector written.

1 Read: One or more interrupt vectors have been written to interrupt queue.

Write: Clear bit

GMASK Global Interrupt Mask Register

Access : read/write
Address : 0F4_H
Reset Value : FFFFFFFF_H

| | | | | | | | | | | | | | | | |
|-------|---|---|---|---|---|---|----|----|----|----|----|----|----|------|----|
| 31 | | | | | | | | | | | | | 17 | 16 | |
| INTOF | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | LINT | IF |
| | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | Q8 | Q7 | Q6 | Q5 | Q4 | Q3 | Q2 | Q1 | Q0 |

Each bit in this register mask the interrupts, which are flagged in register GISTA/GIACK.

- INTOF** Mask Interrupt Overflow
This bit masks the interrupt overflow interrupt.
- LINT** Local Bus Interrupt
This bit masks bridging of interrupt from the local bus to the PCI bus.
0 Bridging of $\overline{\text{LINT}}$ to $\overline{\text{INTA}}$ enabled.
1 Bridging of $\overline{\text{LINT}}$ to $\overline{\text{INTA}}$ disabled.
- IF** Interrupt FIFO
This bit masks the internal mailbox/layer one interrupt FIFO.
0 IF interrupt is enabled.
1 IF interrupt is disabled.
- Q8..Q0** Mask Interrupt Queue 8..0
Each of the bits Q8..Q0 masks an interrupt, which will be asserted, when an interrupt vector has been written to the corresponding interrupt queue 8..0. Masking an interrupt does not suppress generation of the interrupt vector itself.
0 Enable interrupt, when interrupt vector has been written to selected interrupt queue.
1 Mask (Disable) interrupt, when interrupt vector has been written to selected interrupt queue.

8.2.3 PCI and Local Bus Slave Register Set

FCONF Framer and FDL Configuration Register

Access : read/write
Address : 100_H (PCI), 00_H (Local Bus)
Reset Value : 8080_H

| | | | | | | | | | | | | | | | |
|-----|----|---|---|---|---|---|---|------|-----|-----|-----|-----|-----|-----|-----|
| 15 | 14 | | | | | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| IIP | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MBID | WSE | BSD | P28 | P18 | P08 | LAE | LME |

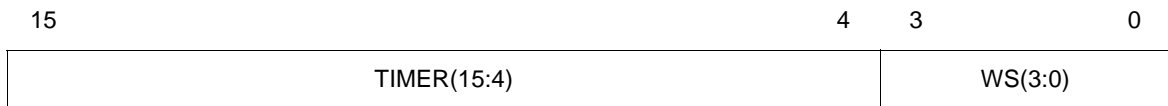
- IIP** Initialization in Progress (Read Only)
After reset (hardware reset or software reset) the internal RAM's are self initialized by the MUNICH256FM. During this time (approx. 250 μs) no other accesses to the device than reading register CONF1 or FCONF are allowed. This bit must be polled until it has been deasserted by the MUNICH256FM.
- 0 Self initialization has finished.
 - 1 Self initialization in progress.
- MBID** Mailbox Interrupt Vector Disable
- 0 Enable generation of mailbox interrupt vectors. As soon as system software on PCI side writes to register MBP2E0 an interrupt vector indicating a mailbox interrupt will be forwarded to the internal interrupt FIFO and can be read by the local CPU.
 - 1 Disable generation of mailbox interrupt vectors.
- WSE** Wait State Enable
This bit enables the wait state controlled master mode.
- 0 $\overline{\text{LRDY}}$ (Intel), $\overline{\text{LDTACK}}$ (Motorola) controlled bus mode.
 - 1 Wait state controlled bus mode. Wait states are defined in register MTIMER.WS.

Register Description

| | |
|----------|--|
| BSD | <p>Byte Swap Disable</p> <p>This bit disables byte swapping on 16-bit transfers when the local bus is operated in Motorola master mode.</p> <p>0 Enable byte swap.</p> <p>1 Disable byte swap.</p> |
| P28..P08 | <p>Switch Page 2..0 to 8-bit mode</p> <p>The MUNICH256FM maps three pages of 8 kByte each to the local bus in master mode. Each page accessed from the PCI side can be mapped in 8-bit mode or 16-bit mode. In 8-bit mode the data bits LD(15:8) are unused.</p> <p>0 Set page mode to 16-bit mode.</p> <p>1 Set page mode to 8-bit mode.</p> |
| LAE | <p>Local Bus Arbiter Enable</p> <p>This bit enables the local bus arbiter. In case that the local bus arbiter is enabled the MUNICH256FM will arbitrate for each bus access on the local bus using the arbitration signals. If local bus arbiter functionality is disabled it assumes bus ownership and does not arbitrate for the local bus.</p> <p>0 Disable the local bus arbiter.</p> <p>1 Enable the local bus arbiter.</p> |
| LME | <p>Local Bus Master Enable</p> <p>This bit enables the local bus master functionality. As long as the local bus master functionality is disabled the MUNICH256FM can be accessed from the local bus as slave only.</p> <p>0 Disable Local Bus Master.</p> <p>1 Enable Local Bus Master.</p> |

MTIMER
Master Local Bus Timer Register

Access : read/write
Address : 104_H (PCI), 02_H (Local Bus)
Reset Value : 0000_H



TIMER Local Bus Latency Timer
 TIMER*16 determines the time in clock cycles the MUNICH256FM holds the local bus as bus master after it was granted the bus. It holds the bus as long as the first transaction is in progress or the latency timer is counting. In case that the MUNICH256FM shall release the bus after it each transaction the latency TIMER value must be set to zero.

WS Wait State Timer
 The value of this register determines the time in clock cycles the MUNICH256FM asserts $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ (Intel Mode) respectively $\overline{\text{LDS}}$ (Motorola Bus Mode). See also FCONF.WSE.

INTCTRL
Interrupt Control Register

Access : read/write
Address : 108_H (PCI), 04_H (Local Bus)
Reset Value : 0001_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|----|----|------|----|
| 15 | | | | | | | | | | | | | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ID | IP | CLIQ | IM |

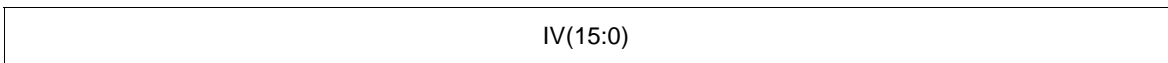
- ID** Interrupt Direction
This pin determines the direction of the interrupt pin $\overline{\text{LINT}}$.
0 $\overline{\text{LINT}}$ is output.
1 $\overline{\text{LINT}}$ is input.
- IP** Interrupt Polarity
0 $\overline{\text{LINT}}$ is active low.
1 $\overline{\text{LINT}}$ is active high.
- CLIQ** Clear Interrupt Queue
Setting this bit will clear the internal interrupt FIFO. This effects all interrupts of facility data link, framer and mailbox interrupts to the local bus.
0 No action
1 Clear interrupt FIFO.
- IM** Interrupt Mask
This bit masks assertion of the pin $\overline{\text{LINT}}$ when interrupts are stored in the internal interrupt FIFO. If the interrupt direction bit is set to output mode interrupt are flagged at interrupt pin $\overline{\text{LINT}}$. If the interrupt direction is set to input mode interrupts are flagged at pin $\overline{\text{LINT}}$.
0 Enable assertion of interrupt pin $\overline{\text{LINT}}$.
1 Disable assertion of interrupt pin $\overline{\text{LINT}}$.

INTFIFO
Interrupt FIFO

Access : read
Address : 10C_H (PCI), 06_H (Local Bus)
Reset Value : FFFF_H

15

0



IV **Interrupt Vector**

After the MUNICH256FM asserted interrupt pin $\overline{\text{LINT}}$ on the local bus side, this bit field contains an interrupt vector containing interrupt information. Please refer to section "Layer One Interrupts" on page 136 for a detailed description of interrupt vector contents.

TREGSEL Transmit T1/E1 Framer Port & Register Select

Access : read/write
Address : 110_H (PCI), 08_H (Local Bus)
Reset Value : 0000_H

| | | | | | | | | | | |
|----|-----|----|-----------|---|---|-----|---|---|---|-----------|
| 15 | 14 | 12 | 8 | 7 | 3 | 0 | | | | |
| 0 | AIP | 0 | PORT(4:0) | | | AIA | 0 | 0 | 0 | ADDR(3:0) |

Note: This register is an indirect access register, which must be programmed before accessing the register TDATA.

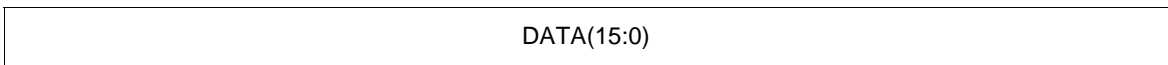
- AIP** Auto Increment Port
This bit enables the auto increment function of bit field PORT. Each read/write access to register TDATA increments PORT. This allows to program multiple, consecutive ports without accessing TREGSEL again.
0 Disable auto increment function.
1 Enable auto increment function.
- PORT** Port Select
This bit field selects the port number, which can be accessed via register TDATA.
0..27 Port Number.
- AIA** Auto Increment Address
This bit enables the auto increment function of bit field ADDR. Each read/write access to register TDATA increments ADDR. This allows to program multiple, consecutive registers without accessing TREGSEL again.
0 Disable auto increment function.
1 Enable auto increment function.
- ADDR** Register Address
This bit field selects the register address of the transmit framer, which can be accessed via register TDATA.
0_H..F_H Register address.

TDATA
Transmit T1/E1 Framer Data Register

Access : read/write
Address : 114_H (PCI), 0A_H (Local Bus)
Reset Value : 0000_H

15

0

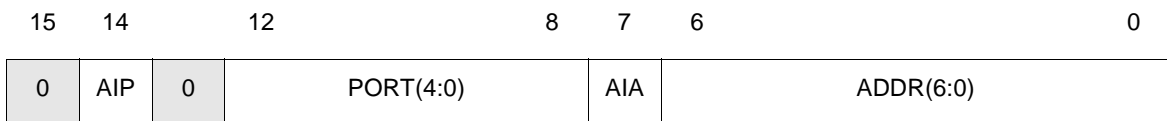


Note: Effected port and address is selected via register TREGSEL. All settings in this register affect the selected port only.

DATA Data register
The transmit framer data register assigns a value to the transmit framer of port TREGSEL.PORT and the register selected via bit field TREGSEL.ADDR. Read/write operation depends on the selected register.

RREGSEL
Receive T1/E1 Framer Port & Register Select

Access : read/write
Address : 118_H (PCI), 0C_H (Local Bus)
Reset Value : 0000_H



Note: This register is an indirect access register, which must be programmed before accessing the register RDATA.

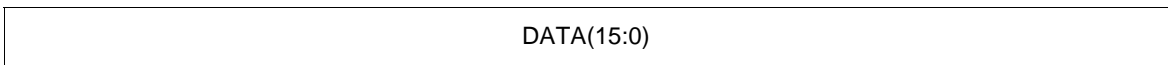
- AIP** Auto Increment Port
- This bit enables the auto increment function of bit field PORT. Each read/write access to register RDATA increments PORT. This allows to program multiple, consecutive ports without accessing RREGSEL again.
- 0 Disable auto increment function.
1 Enable auto increment function.
- PORT** Port Select
- This bit field selects the port number, which can be accessed via register RDATA.
- 0..27 Port Number.
- AIA** Auto Increment Address
- This bit enables the auto increment function of bit field ADDR. Each read/write access to register RDATA increments ADDR. This allows to program multiple, consecutive registers without accessing RREGSEL again.
- 0 Disable auto increment function.
1 Enable auto increment function.
- ADDR** Register Address
- This bit field selects the register address of the transmit framer, which can be accessed via register RDATA.
- 0_H..7F_H Register address.

RDATA
Receive T1/E1 Frammer Data Register

Access : read/write
Address : 11C_H (PCI), 0E_H (Local Bus)
Reset Value : 0000_H

15

0



Note: Effected port and address is selected via register RREGSEL. All settings in this register affect the selected port only.

DATA Data register
The receive framer data register assigns a value to the receive framer of port RREGSEL.PORT and the register selected via bit field RREGSEL.ADDR. Read/write operation depends on the selected register.

FREGSEL
FDL Port & Register Select

Access : read/write
Address : 120_H (PCI), 10_H (Local Bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | |
|-----|---|---|-----------|--|--|--|-----|---|---|-----------|--|---|--|--|---|
| 15 | | | 12 | | | | | 8 | 7 | | | 4 | | | 0 |
| AIP | 0 | 0 | PORT(4:0) | | | | AIA | 0 | 0 | ADDR(4:0) | | | | | |

Note: This register is an indirect access register which must be programmed before accessing the register FDATA.

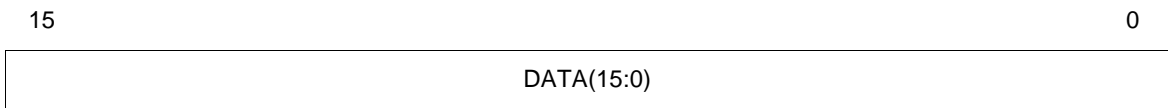
- AIP** Auto Increment Port
This bit enables the auto increment function of bit field PORT. Each read/write access to register FDATA increments PORT. This allows to program multiple, consecutive ports without accessing FREGSEL again.
0 Disable auto increment function.
1 Enable auto increment function.
- PORT** Port Select
This bit field selects the port number, which can be accessed via register FDATA.
0..27 Port Number for T1/E1.
28 Far End Alarm and Control Channel (DS3)
29 C-bit parity path maintenance data link channel (DS3)
- AIA** Auto Increment Address
This bit enables the auto increment function of bit field ADDR. Each read/write access to register FDATA increments ADDR. This allows to program multiple, consecutive registers without accessing FREGSEL again.
0 Disable auto increment function.
1 Enable auto increment function.

Register Description

ADDR Register Address
This bit field selects the register address of the facility data link channel, which can be accessed via register FDATA.
0_H..1F_H Register address.

FDATA
FDL Data Register

Access : read/write
 Address : 124_H (PCI), 12_H (Local Bus)
 Reset Value : 0000_H



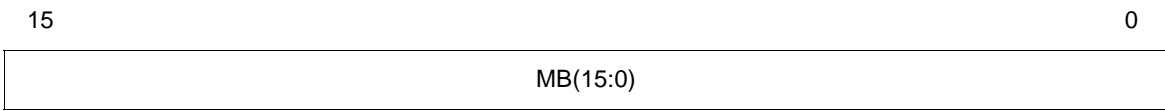
Note: Effected port and address is selected via register FREGSEL. All settings in this register affect the selected port only.

DATA Data register

The FDL data register assigns a value to the facility data link controller of port FREGSEL.PORT and the register selected via bit field FREGSEL.ADDR. Read/write operation depends on the selected register.

MBE2P0
Mailbox Local Bus to PCI Command Register

Access : read/write
 Address : 140_H (PCI), 20_H (Local Bus)
 Reset Value : 0000_H

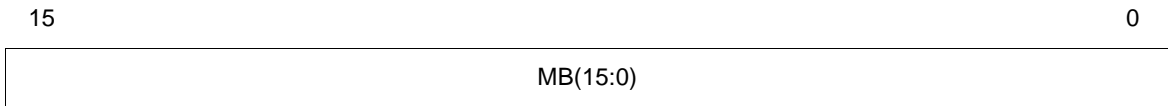


MB Mailbox Data register

This register can be written and read from local bus side. From PCI side this register should be used as read only in order to allow stable interprocessor communication. Write access to this register results in mailbox interrupt vectors on local bus side to the internal interrupt FIFO when FCONF.MBID is set to '0'.

MBE2P1-7
Mailbox Local Bus to PCI Data Register 1-7

Access : read/write
 Address : 144_H-15C_H (PCI), 22_H-2E_H (Local Bus)
 Reset Value : 0000_H



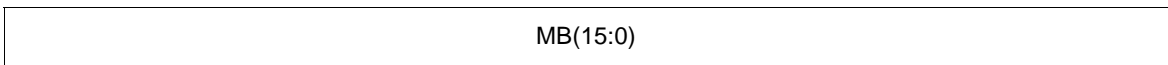
MB Mailbox Data register
 This register can be written and read from local bus side. From PCI side this register should be used as read only in order to allow stable interprocessor communication.

MBP2E0
Mailbox PCI to Local Bus Status Register

Access : read/write
 Address : 160_H (PCI), 30_H (Local Bus)
 Reset Value : 0000_H

15

0



MB Mailbox Status Register

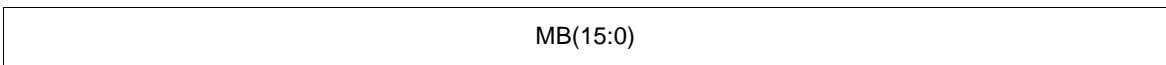
This register can be written and read from PCI side. From local bus side this register should be used as read only in order to allow stable interprocessor communication. Write access to this register results in mailbox interrupt vectors to PCI side when CONF1.MBIM is set to '0'.

MBP2E1-7
Mailbox PCI to Local Bus Data Register 1-7

Access : read/write
 Address : 164_H-17C_H (PCI), 32_H-3E_H (Local Bus)
 Reset Value : 0000_H

15

0



MB Mailbox Data Register

This register can be written and read from PCI side. From local bus side this register should be used as read only in order to allow stable interprocessor communication.

8.2.3.1 M13 Transmit Registers

D3CLKCS DS3 Clock Configuration and Status Register

Access : read/write
Address : 180_H (PCI), 40_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|-----|-----|-----|-----|------|------|------|
| 15 | | | | | | | | | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RCA | TCA | RRX | RTX | T2RL | R2TL | TXLT |

- RCA** Receive Clock Activity
This bit monitors the receive clock activity (RC44).
0 No receive DS3 clock since last read of this register. This bit is set to '0' approx. 125 μs after the last active clock was detected.
1 At least one receive DS3 clock since last read of this register.
- TCA** Transmit Clock Activity
This bit monitors the transmit clock activity (TC44).
0 No transmit DS3 clock since last read of this register. This bit is set to '0' approx. 125 μs after the last active clock was detected.
1 At least one transmit DS3 clock since last read of this register.
- RRX** Reset Receiver Clock Unit
This bit resets the receivers clock unit.
0 Normal operation.
1 Reset DS3 receiver clock unit. This bit is self clearing.
- RTX** Reset Transmitter Clock Unit
This bit resets the transmitters clock unit.
0 Normal operation.
1 Reset DS3 transmitter clock unit. This bit is self clearing.

Register Description

| | |
|------|--|
| T2RL | <p>Transmit to Receive Loop (Local DS3 Loopback)</p> <p>This bit enables the local DS3 loop where the outgoing DS3 bit stream is mirrored to the DS3 input.</p> <p>0 Disable local loop.</p> <p>1 Enable local loop.</p> |
| R2TL | <p>Receive to Transmit Loop (Remote DS3 Loopback)</p> <p>This bit enables the remote DS3 line loop where the complete incoming DS3 bit stream is mirrored to the transmitter.</p> <p>0 Disable remote loop.</p> <p>1 Enable remote loop.</p> |
| TXLT | <p>Transmit Loop Timing Mode</p> <p>This bit enables DS3 looped timing where the transmitter uses the receivers DS3 input clock.</p> <p>0 Disable looped timing.</p> <p>1 Enabled looped timing.</p> |

TUCLKC
Test Unit Clock Configuration Register

Access : read/write
Address : 184_H (PCI), 42_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------|-----|
| 15 | | | | | | | | | | | | | | | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RTUR | TUL |

RTUR **Reset Test Unit Receiver**
This bit resets the test unit receiver.
0 Normal operation.
1 Reset Receiver (automatically removed). This bit is self clearing.

TUL **Test Unit Transmit to Receive Loop**
This bit switches a local loop from the test unit transmitter to the test unit receiver. While operating in loop mode the test unit is operated with TC44.
0 Normal operation.
1 Test unit transmitter output connected to test unit receiver input.

D3TCFG
DS3 Transmit Configuration Register

Access : read/write
Address : 188_H (PCI), 44_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|-----|------|-----|-----|------|----------|-----|-----|---|
| 15 | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FAM | ITCK | ITD | UTD | AISC | LPC(1:0) | FPL | CBP | |

- FAM** TOVHSYN Mode
 This bit switches between input mode and output mode of the signal pin TOVHSYN. If TOVHSYN is operated in input mode it marks the position of the X-bit. Therefor the outgoing DS3 frame is aligned to TOVHSYN. If TOVHSYN is switched to output mode TOVHSYN is asserted when the X-bit needs to be inserted via the transmit overhead interface.
 0 TOVHSYN switched to input.
 1 TOVHSYN switched to output.
- ITCK** Invert Transmit Clock
 This bit sets the clock edge for data transmission.
 0 Update transmit data on the rising edge of transmit clock.
 1 Update transmit data on the falling edge of transmit clock.
- ITD** Invert Transmit Data
 This bit enables inversion of transmit data.
 0 Transmit data is logic high (not inverted).
 1 Transmit data is logic low (inverted).
- UTD** Unipolar data mode
 This bit sets the port mode to dual-rail mode or unipolar mode.
 0 B3ZS (dual rail data)
 1 Unipolar mode (single rail data)

Register Description

| | |
|------|---|
| AISC | <p>AIS Code Type</p> <p>This bit field sets the AIS code.</p> <p>0 Set AIS to '1010...' between overhead bits, C-bits all '0's, X-bits all '1's (standard)</p> <p>1 Set AIS to unframed all '1's (non-standard).</p> |
| LPC | <p>Loopback Code.</p> <p>This bit field selects the C-bit which will be inverted when loopback requests are transmitted.</p> <p>00 Invert 1st C-bit.</p> <p>01 Invert 2nd C-bit.</p> <p>10 Invert 3rd C-bit.</p> |
| FPL | <p>Full Payload Mode</p> <p>This bit enables the M23 multiplex operation or the full payload rate format.</p> <p>0 Enable M23 multiplex operation. Payload is formed by interleaving 7 asynchronous DS2 tributaries</p> <p>1 Enable full payload rate format. The payload is one single, high speed data stream without stuffing.</p> |
| CBP | <p>C-bit parity mode</p> <p>This bit enables M13 asynchronous mode or C-bit parity mode.</p> <p>0 M13 asynchronous mode</p> <p>1 C-bit parity mode</p> |

D3TCOM
DS3 Transmit Command Register

Access : read/write
Address : 18C_H (PCI), 46_H (Local bus)
Reset Value : 0070_H

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|------|-------------------|-------|-------|-------|------|---|
| 15 | | | | | | | | | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TAIC | TN _r B | TXBIT | SIDLE | SAISA | SAIS | 0 |

- TAIC** Transmitted AIC-bit
This bit sets the value to be transmitted in the DS3 overhead bit of block 3, subframe 1. This function is available in C-pit parity format only.
0 AIC-bit = '0'
1 AIC-bit = '1'
- TN_rB** Transmitted N_r-bit
This bit sets the value to be transmitted in the DS3 overhead bit of block 5, subframe 1. This function is available in C-pit parity format only.
0 N_r-bit = '0'
1 N_r-bit = '1'
- TXBIT** Transmitted X-bits
This bit sets the value to be transmitted in the DS3 overhead bit of block 1, subframes 1 and 2.
TXBIT is synchronized to the M23 multiframe. Both X-bits in a multiframe are guaranteed identical. Software should limit changes to maximum of 1 per second.
0 X-bit = '0'
1 X-bit = '1'
- SIDLE** Send DS3 Idle Code
This bit enables transmission of the DS3 idle code ('1010' between overhead bits, X-bits all '1's, C-bits all '0's).
0 Normal operation.
1 Send DS3 idle code.

Register Description

| | |
|-------|--|
| SAISA | Send AIS in DS3 output and on DS3 loop) This bit enables transmission of AIS on the DS3 output. If the DS3 is additionally switched to local DS3 loopback mode the DS3 signal including AIS is mirrored to the receiver. The AIS code transmitted depends on D3TCFG.AISC. 0 Normal operation. 1 Enable transmission of AIS. |
| SAIS | Send AIS at DS3 output This bit enables transmission of AIS on the DS3 output. If the DS3 signal is switched into local DS3 loopback mode the DS3 signal without AIS code is mirrored to the DS3 receiver. The AIS code transmitted depends on D3TCFG.AISC. 0 Normal operation. 1 Enable transmission of AIS. |

D3TLPB DS3 Transmit Remote DS2 Loopback Register

Access : read/write
Address : 190_H (PCI), 48_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|----------|---|
| 15 | | | | | | | | 6 | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LPB(6:0) | |

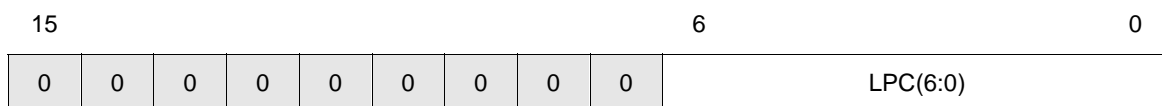
LPB Remote DS2 Loopback

Setting LPB(x) enables the remote DS2 loopback of tributary x. In this mode the demultiplexed DS2 tributary is internally looped and multiplexed into the outgoing DS3 signal.

| | |
|---|--|
| 0 | Normal operation. |
| 1 | Enable remote DS2 loopback of tributary x. |

D3TLPC DS3 Transmit Loopback Code Insertion Register

Access : read/write
 Address : 194_H (PCI), 4A_H (Local bus)
 Reset Value : 0000_H



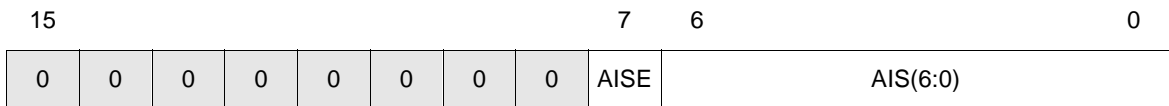
LPC Send Loopback

Setting LPC(x) enables transmission of the loopback code in tributary x of the DS3 signal. The loopback code inserted depends on D3TCFG.LPC.

| | |
|---|--|
| 0 | Normal operation. |
| 1 | Enable transmission of loopback code in tributary x. |

D3TAIS
DS3 Transmit AIS Insertion Register

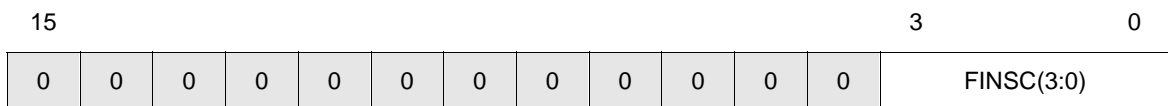
Access : read/write
Address : 198_H (PCI), 4C_H (Local bus)
Reset Value : 0000_H



- AISE** AIS Error Insertion
 Toggling this bit inserts one '0' in all DS3 tributaries which transmit AIS.
- AIS** Send DS2 Alarm Indication Signal
 Setting AIS(x) enables insertion of the DS2 alarm indication signal in the outgoing tributary x of the DS3 signal. AIS is an all '1' signal.
- 0 Normal operation
 1 Enable transmission of AIS in tributary x.

D3TFINS
DS3 Transmit Fault Insertion Control Register

Access : read/write
Address : 19C_H (PCI), 4E_H (Local bus)
Reset Value : 0000_H



- FINSC** Fault Insertion Code.
- Fault insertion is service affecting and is intended for testing only. Codes are not self clearing, i.e. errors are continuously generated as indicated until bit cleared. A single FEBE, P, CP, or code violation is guaranteed to be inserted if the respective code is written and then immediately cleared.
- 0 Normal operation (no fault insertion)
 - 1 Insert FEBE event every multiframe (106 μsec).
 - 2 Insert P-bit errors every 2nd multiframe (212 μsec).
 - 3 Insert CP-bit errors every 2nd multiframe (212 μsec).
 - 4 Insert 4 F-bit errors/multiframe (satisfies 3 out of 15 threshold trigger).
 - 5 Insert 5 F-bit errors/multiframe (satisfies 3 out of 7 threshold trigger).
 - 6 Insert 3 M-bit errors/multiframe (caution: receiver can frame on emulator).
 - 7 Force DS3 output to all '0's.
 - 8 Insert B3ZS violation/multiframe (violation of alternate polarity rule).
 - 9 Insert 3 zero string/multiframe (B3ZS code word suppressed)

D3TTUC
DS3 Transmit Test Unit Control Register

Access : read/write
Address : 1A0_H (PCI), 50_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|----|------------|--|--|------------|--|---|------|---|---|---|
| 15 | | | | | | | | | 7 | 6 | | | | | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EN | TUDS2(2:0) | | | TUDS1(1:0) | | | TUIM | | | |

- EN** Enable Test Unit Insertion
Setting this bit enables insertion of the test unit data.
0 Normal operation
1 Enable insertion of test unit data.
- TUDS2** Test Unit DS2 Group
This bit field selects the DS2 group the test unit is attached to. Only valid if TUIM is 10_B, 01_B or 00_B.
0..6 Selects DS2 group 0..6.
- TUDS1** Test Unit DS1 Tributary
This bit field selects the DS1 tributary the test unit is attached to. Only valid if TUIM is 00_B. The DS2 group is selected via TUDS2.
0..3 DS1/E1 tributary
- TUIM** Bit Error Rate Test Unit (TU) Insertion Mode
This bit field selects the interface the test unit is attached to.
00_B Insert test stream into DS1/E1 tributary (unframed)
01_B Insert test stream into DS2 tributary (unframed, bypass M12)
10_B Insert test stream into DS2 payload (framed)
11_B Insert test stream into DS3 payload (framed)

D3TSDL
DS3 Transmit Spare Data Link Register

Access : read/write
Address : 1A4_H (PCI), 52_H (Local bus)
Reset Value : 01FF_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|------|------|------|------|------|------|------|------|------|
| 15 | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DL77 | DL75 | DL73 | DL67 | DL65 | DL63 | DL27 | DL25 | DL23 |

Multiframe buffer for spare DL bits transmitted in blocks 3, 5, and 7 of subframes 2, 6, and 7. If enabled, the M13 will generate an interrupt every multiframe to request a refresh of this register. The software must write these registers within 106 μsec to avoid an underrun.

DL(S)(B) Overhead bit for block B of subframe S
These bits store the DL bits to be transmitted in blocks 3, 5, and 7 of subframes 2, 6, and 7. If enabled, the M13 will generate an interrupt every multiframe to request a refresh of this register. The software must write these registers within 106 μsec to avoid an underrun.

D3RCFG
DS3 Receive Configuration Register

Access : read/write
Address : 1C0_H (PCI), 60_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|-----|---|---|---|-----|------|-----|------|---|------|-----|------|-----|------|-----|-----|---|
| 15 | | | | | 11 | 10 | 9 | 8 | | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| CVM | 0 | 0 | 0 | IVM | STTM | ECM | FEBM | 0 | AISX | MFM | MDIS | FFM | IRCK | IRD | URD | |

Note: M13 mode, Full payload mode, loopback code, and AIS mode are controlled by bits CBP, FPL, LPC, and AISX in register DS3 transmit configuration register D3TCFG.

- CVM** B3ZS Code Word (“00V” or “10V” Acceptance Condition)
This bit selects the B3ZS violations alternate polarity to maintain line balance.
0 Convert all B3ZS codeword patterns to “000” regardless of polarity.
1 Convert codeword only if alternate violation polarity rule is satisfied.
- IVM** Interrupt Vector Mode
This bit selects the interrupt vector mode.
0 Interrupt vector flags are set when corresponding condition has changed.
1 Interrupt vector flags contain actual status of condition.
- STTM** Select Transmit Tributary Monitoring for receive test unit
This bit selects the T1/E1 tributary observed by the test unit receiver. The test unit can be connected to the upstream T1/E1 tributary (T1/E1 tributary going towards the DS3 interface) or to the downstream T1/E1 tributary (T1/E1 tributary coming from the DS3 interface).
0 Monitor downstream T1/E1 tributary.
1 Monitor upstream T1/E1 tributary.

Register Description

| | |
|-------------|--|
| ECM | <p>Error Counter Mode</p> <p>DS3 errors are counted in background and copied to foreground (error counter registers) when condition selected via ECM is met.</p> <p>0 Counter values are copied to foreground when copy command is executed. See also register DS3COM.</p> <p>1 The counter values are copied to the foreground register in one second intervals. At the same time the background registers are reset to zero. This operation is synchronous with the periodic one second interrupt which alerts software to read the register.</p> |
| FEBM | <p>Far End Block Error (FEBE) Mode</p> <p>This bit selects the event which leads to FEBE indication. It is available in C-bit parity mode only.</p> <p>0 Receive multiframe parity error.</p> <p>1 Receive multiframe parity error or framing error.</p> |
| AISX | <p>AIS X-bit Check Disable</p> <p>This bit disables checking of the X-bit for AIS and idle detection.</p> <p>0 Check X-bit.</p> <p>1 Disable check of X-bit.</p> |
| MFM | <p>Multiframe Framing Mode</p> <p>This bit selects the M-bit error condition which triggers the DS3 framer to start a new frame search. To enable reframing in case of M-bit errors MDIS must be set to '0'.</p> <p>0 Start new F-frame search if M-bit errors are detected in two out of four consecutive M-frames.</p> <p>1 Start new F-frame search if M-bit errors are detected in three out of four consecutive M-frames.</p> |
| MDIS | <p>Multiframe Reframe Disable</p> <p>This bit disables reframing due to M-bit errors.</p> <p>0 Enable reframe due to M-bit errors.</p> <p>1 Disable reframe due to M-bit errors.</p> |

Register Description

| | |
|------|---|
| FFM | F Framing Mode This bit selects the F-bit error condition which triggers the DS3 framer to start a new frame search. 0 A new frame search is started when 3 out of 8 contiguous F-bits are in error. 1 A new frame search is started when 3 out of 16 contiguous F-bits are in error. |
| IRCK | Invert Receive Clock This bit sets the clock edge for data sampling. 0 Sample data on the rising edge of receive clock. 1 Sample data on the falling edge of receive clock. |
| IRD | Invert Receive Data This bit enables inversion of receive data. 0 Receive data is logic high (not inverted). 1 Receive data is logic low (inverted). |
| URD | Unipolar Receive Data This bit sets the port mode to dual-rail mode or unipolar mode. 0 B3ZS (dual rail data input) 1 Unipolar mode (single rail data input) |

D3RCOM
DS3 Receive Command Register

Access : read/write
Address : 1C4_H (PCI), 62_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|------|-----|------|-----|-----|
| 15 | | | | | | | | | | | | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | C3NC | C3C | CNCA | CCA | FRS |

C3NC Copy DS3 Error Counters
Values of DS3 background registers are copied to foreground. Background registers are NOT cleared. Command is self clearing and completes before next register access is possible i.e. software can write command and then immediately read the counters without starting a delay timer.

Note: Usage of this function in not recommend in 'One Second' error counter mode (D3RCFG.ECM = '1').

- 0 No operation.
- 1 Copy background counters to foreground.

C3C Copy and Clear DS3 Error Counters
Values of DS3 background registers are copied to foreground. Background registers are cleared. Command is self clearing and completes before next register access is possible i.e. software can write command and then immediately read the counters without starting a delay timer.

- 0 No operation.
- 1 Copy background counters to foreground. Clear background counters.

Note: Usage of this function in not recommend in 'One Second' error counter mode (D3RCFG.ECM = '1').

CCNA Copy Error Counters
Only valid for counters which are not operating in 'One Second' error counter mode. Values of DS2 and DS3 background registers are copied to foreground. Background registers are NOT cleared. Command is self

Register Description

clearing and completes before next register access is possible i.e. software can write command and then immediately read the counters without starting a delay timer.

0 No operation.

1 Copy background counters to foreground.

CCA

Copy and Clear DS2/DS3 Error Counters

Only valid for counters which are not operating in 'One Second' error counter mode. Values of DS2 and DS3 background registers are copied to foreground. Background registers are cleared. Command is self clearing and completes before next register access is possible i.e. software can write command and then immediately read the counters without starting a delay timer.

0 No operation.

1 Copy background counters to foreground. Clear background counters.

FRS

Force Resynchronization

This bit enables a new frame search on the DS3 input. The command is self clearing after frame search has begun.

0 Normal operation.

1 Force new frame search.

D3RIMSK
DS3 Receive Interrupt Mask Register

Access : read/write
Address : 1C8_H (PCI), 64_H (Local bus)
Reset Value : 1FFF_H

| | | | | | | | | | | | | | | | |
|----|---|---|------|------|------|------|-----|----------------|-----|------|-------|------|------|------|-----|
| 15 | | | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | CLKS | RSDL | TSDL | LPCS | SEC | N _r | AIC | XBIT | IDLES | AISS | REDS | LOSS | FAS |

This register provides the interrupt mask for DS3 status interrupts and DS3 loopback code interrupts. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see "Layer One Interrupts" on page 136.

The following definition applies:

- 1 The corresponding interrupt vector will not be generated by the device.
- 0 The corresponding interrupt vector will be generated.

- RSDL Mask 'Receive Spare Data Link Transfer Buffer Full'
- TSDL Mask 'Transmit Spare Data Link Transfer Buffer Empty'
- LPCS Mask 'Loopback Code Status' (flagged in D3RLPCS)
- SEC Mask '1 Second Interrupt'

- CLKS Mask 'DS3 Clock Status'
- N_r Mask 'N_r-bit Image' (C-bit parity mode only)
- AIC Mask 'AIC-bit Image' (C-bit parity mode)
- XBIT Mask 'X-bit Image'
- IDLES Mask 'DS3 Idle Signal State'
- AISS Mask 'DS3 Alarm Indication Signal State'
- REDS Mask 'DS3 Red Alarm State'
- LOSS Mask 'DS3 Input Signal State'
- FAS Mask 'Frame Alignment State'

D3RESIM
DS3 Receive Error Simulation Register

Access : read/write
Address : 1CC_H (PCI), 66_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|------|---|------------|---|
| 15 | | | | | | | | | | | 4 | | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FTMR | 0 | ESIMC(2:0) | |

FTMR Fast Timer
This bit enables alarm timer test function (manufacturing test only).
0 Normal Operation
1 Test Operation
 DS3 RED/AIS/Idle timer period reduced by 56.
 DS2 READ/AIS timer period reduced by 24.
 Second interrupt period reduced to 140 μsec

ESIMC Error Simulation Code
This bit enables error simulation. During error simulation the device generates error interrupts and error status messages. Nevertheless the service is not affected.
0 Normal operation (no error simulation).
1 Simulate one F-bit error/multiframe (106 μsec).
2 Simulate M-bit error in every other multiframe.
3 Simulate FEBE event/multiframe (106 μsec).
4 Simulate P/CP event/multiframe (106 μsec).
5 Simulate Loss of DS3 input (all zeros).
6 Simulate B3ZS code violations.
7 Simulate Loss of Receive Clock

D3RTUC
DS3 Receive Test Unit Control Register

Access : read/write
Address : 1D0_H (PCI), 68_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|----|------------|--|--|------------|--|---|------|---|---|---|
| 15 | | | | | | | | | 7 | 6 | | | | | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | EN | TUDS2(2:0) | | | TUDS1(1:0) | | | TURM | | | |

- EN** Enable Test Unit Receive Clock
This bit enables the receive clock of the test unit. The clock speed is dependent on the selected test mode.
0 Receive clock disabled.
1 Receive clock enabled.
- TUDS2** Test Unit DS2 Group
This bit field selects the DS2 group the test unit is attached to. Only valid if TURM is 10_B, 01_B, or 00_B.
0..6 Selects DS2 group 0..6.
- TUDS1** Test Unit DS1/E1 Tributary
This bit field selects the DS1/E1 tributary the test unit is attached to. Only valid if TURM is 00_B. The DS2 group is selected via TUDS2.
0..3 DS1/E1 tributary
- TURM** Test Unit Receive Mode
This bit field selects the interface the test unit is attached to.
00_B DS1/E1 tributary (unframed)
01_B DS2 tributary (unframed, bypass M12)
10_B DS2 payload (framed)
11_B DS3 payload (framed)

D3RSTAT
DS3 Receive Status Register

Access : read
Address : 1D4_H (PCI), 6A_H (Local bus)
Reset Value : 0841_H

| | | | | | | | | | | | | | | | |
|----|------|------|------|------|------|-----|------------------------|-----|------|-------|------|------|------|------|-----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | LRXC | LTXC | RSDL | TSDL | LPCD | SEC | N _r AICC | AIC | XBIT | IDLES | AISS | REDS | LOSS | COFA | FAS |

Each bit in the DS3 framer receive status register declares a specific condition dependent on the selected modes. The following convention applies to the individual bits:

- 0 The named status is not or no longer existing.
- 1 The named status is currently effective.

Except for COFA every bit can be used to generate a DS3 interrupt vector. See also register D3RIMSK which describes how to enable/disable interrupt vector generation and refer to the description of DS3 framer interrupts on page "Layer One Interrupts" on page 136.

- LRXC** Loss of Receive DS3 Clock
This bit indicates loss of DS3 receive clock.
- LTXC** Loss of Transmit DS3 Clock
This bit indicates loss of DS3 transmit clock.
- RSDL** Receive Spare Data Link Buffer Full
This bit indicates that the spare data link receive buffer (register D3RSDL) is full.
- TSDL** Transmit Spare Data Link Buffer Empty
This bit indicates that the spare data link transmit buffer (register D3TSDL) is empty.
- LPCD** Loopback Code Detected
This bit indicates a changes in register D3RLPCS.

Register Description

| | |
|----------------------|--|
| SEC | <p>1 Second Flag</p> <p>This bit toggles every second synchronously with the one second interrupt. It can be used by software to synchronize 1 second events when the 'One second interrupt' vector is masked.</p> |
| N _r /AICC | <p>N_r-bit Image (C-bit parity format only)</p> <p>This bit contains an image of the DS3 frame overhead bit in block 5 of subframe 1. It is updated only if its state persists for 3 multiframes and DS3 frame is aligned.</p> <p>AIC-bit changed (M13 asynchronous format)</p> <p>This bit indicates a change of the AIC-bit (first C-bit of the first subframe) since the last read of this register.</p> |
| AIC | <p>AIC bit Image (DS3 frame overhead bit in block 3 of subframe 1)</p> <p>This bit contains an image of the DS3 frame overhead bit in block 3 of subframe 1. It is updated only if its state persists for 3 multiframes and DS3 frame is aligned.</p> |
| XBIT | <p>X bit Image (DS3 frame overhead bit in block 1 of subframes 1 and 2)</p> <p>This bit contains an image of the DS3 frame overhead bit in block 1 of subframes 1 and 2. It is updated only if both bits in a DS3 multiframe have the same value, its state persists for at least 3 multiframes and when the DS3 framer is in synchronous state.</p> |
| IDLES | <p>Idle State</p> <p>This bit indicates that the idle pattern (framed ...1100... with C-bits='0' in subframe 3 and X-bits='1') was persistent as per alarm timing parameters defined in register D3RAP. Idle is considered active in a multiframe when fewer than 15 errors are detected. At 10⁻³ error rates, 5 errors per multiframe are typical. The exact time necessary to change the flag could be greater if the FAS flag is not constant. The frame alignment state is integrated by incrementing or decrementing a counter at the end of each multiframe when the FAS flag is set or cleared respectively.</p> |
| AISS | <p>AIS Slarm State.</p> <p>This bit indicates the AIS alarm state. AIS can be a framed '..1010..' pattern with C-bits='0' and X-bits='1' or an unframed all '1' pattern. This is determined by D3TCFG.AISC. AIS is considered active in a multiframe when fewer than 15 errors are detected and is declared when it was persistent as per alarm timing parameters defined in register D3RAP. At 10⁻³ error rates, 5 errors per multiframe are typical. The exact time necessary to change the flag could be greater if the FAS flag is not constant. The frame alignment state is integrated by incrementing or</p> |

Register Description

decrementing a counter at the end of each multiframe when the FAS flag is set or cleared respectively.

REDS Red Slarm State (loss of frame alignment)

This bit indicates that red alarm was persistent as per alarm timing parameter defined in register D3RAP. The red alarm flag nominally changes when loss of frame alignment condition persists for either 32 or 128 multiframes. This is determined by bit D3RCFG.SAIT. The exact time necessary to change the flag could be greater if the FAS flag is not constant. The frame alignment state is integrated by incrementing or decrementing a counter at the end of each multiframe when the FAS flag set or cleared respectively.

LOSS Loss of DS3 Input Signal

This bit indicates that the received DS3 bit stream contained at least 175 consecutive '0's. It is deasserted when 59 '1' bits are detected in 175 clocks (1/3 density). Following removal of LOS, a 10 msec guard timer is started. If a new LOS occurs, the release condition is extended so that the 1/3 density condition must persist for at least 10 msec. This prevents chatter and excessive interrupts.

COFA Change of Frame Alignment.

This bit indicates a change of frame alignment event. It is set when the DS3 framer found a new frame alignment and when the new frame position differs from the expected frame position.

FAS DS3 Frame Alignment State

This bit indicates that the DS3 framer is not aligned.

D3RSDL
DS3 Receive Spare Data Link Register

Access : read
Address : 1DC_H (PCI), 6E_H (Local bus)
Reset Value : 01FF_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|------|------|------|------|------|------|------|------|------|
| 15 | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | DL77 | DL75 | DL73 | DL67 | DL65 | DL63 | DL27 | DL25 | DL23 |

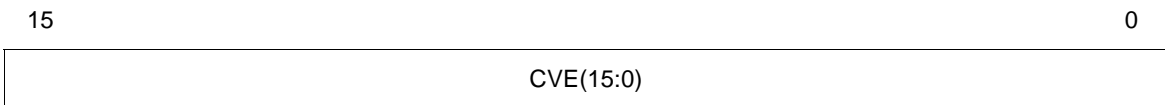
DL(S)(B) **Overhead Bit for Block B of Subframe S**

These bits buffer the spare DL bits received in blocks 3, 5, and 7 of subframes 2, 6, and 7. If enabled, the M13 will generate an interrupt every multiframe to synchronize reading of this register. The register must be read within 106 μsec to avoid an overrun.

D3RCVE

DS3 Receive B3ZS Code Violation Error Counter

Access : read/write
 Address : 1E0_H (PCI), 70_H (Local bus)
 Reset Value : 0000_H

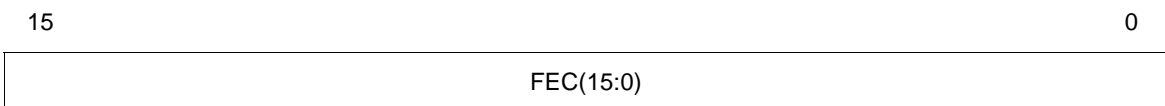


CVE(15:0) B3ZS Code Violation Errors
 Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.
 Count of B3ZS Code Violation errors. The error counter will not be incremented during asynchronous state.

D3RFEC

DS3 Receive Framing Bit Error Counter

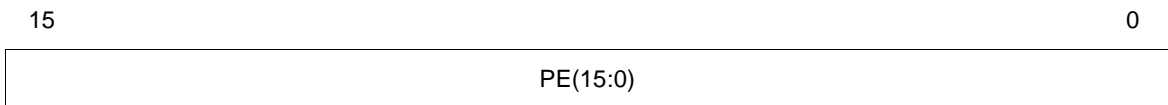
Access : read/write
 Address : 1E4_H (PCI), 72_H (Local bus)
 Reset Value : 0000_H



FEC(15:0) Framing Bit Error Counter
 Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.
 Count of F-bit and M-bit errors. Errors are not counted in out of frame state.

D3RPEC
DS3 Receive Parity Error Counter

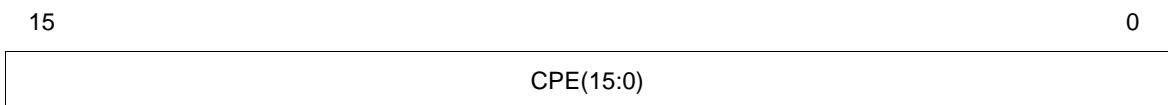
Access : read/write
Address : 1E8_H (PCI), 74_H (Local bus)
Reset Value : 0000_H



PE(15:0) Parity Bit Error Counter
Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.
Count of parity errors (P-bits in DS3 overhead bits). The P-bit is duplicated in the DS3 frame structure but only one error is counted per multiframe. Errors are not counted in out of frame state.

D3RCPEC
DS3 Receive Path Parity Error Counter

Access : read/write
Address : 1EC_H (PCI), 76_H (Local bus)
Reset Value : 0000_H



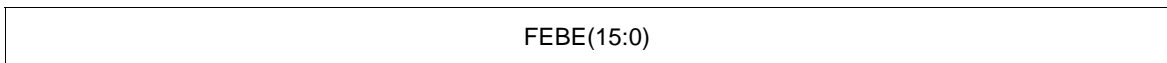
CPE(15:0) Path Parity Error Counter
Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.
Count of path parity errors (CP bits in DS3 C-bit parity overhead bits). CP-bits are triplicated in the DS3 frame structure but only single error maximum is counted per multiframe. Errors are not counted in out of frame state.

D3RFEBEC
DS3 Receive FEBE Error Counter

Access : read/write
Address : 1F0_H (PCI), 78_H (Local bus)
Reset Value : 0000_H

15

0



FEBEC(15:0) FEBE error events

Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.

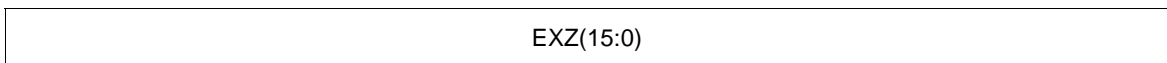
This register counts the occurrence of a received 'not all '1's'. FEBE-bits are triplicated in the DS3 frame structure but only one single error maximum is counted per multiframe. Errors are not counted in out of frame state.

D3REXZ
DS3 Receive Excessive Zeroes Counter

Access : read/write
Address : 1F4_H (PCI), 7A_H (Local bus)
Reset Value : 0000_H

15

0



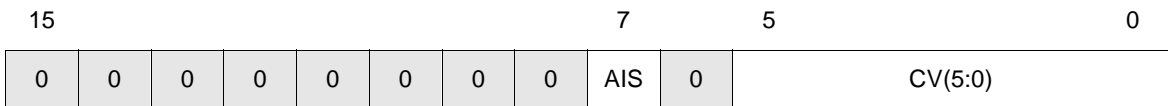
EXZ(15:0) Exzessive Zeroes

Error counter mode (Clear on Read or Errored Second) depends on register D3RCFG.ECM.

Violations are 3 zero strings. The error counter will not be incremented during asynchronous state.

D3RAP
DS3 Alarm Parameters

Access : read/write
Address : 1F8_H (PCI), 7C_H (Local bus)
Reset Value : 0000_H



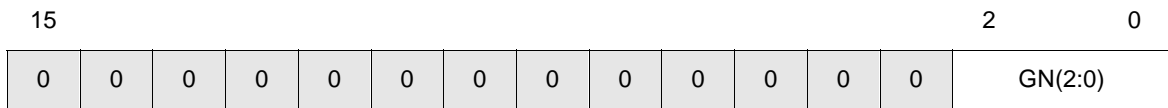
- AIS** AIS criteria
This bits sets the error rate for AIS detection. Declaration of AIS depends on value defined in bit field CV.
- 0 AIS is recognized when the alarm indication signal is received with less than 8 errors per multiframe.
 - 1 AIS is recognized when the alarm indication signal is received with less than 15 errors per multiframe.
- CV** Counter Value
This bit specifies the number of frames when the MUNICH256FM declares AIS, RED or Idle.
0..63 Counter Value.

8.2.3.2 DS2 Control and Status Registers

D2TSEL

DS2 Transmit Group Select Register

Access : read/write
 Address : 200_H (PCI), 80_H (Local bus)
 Reset Value : 0000_H



Note: This register is an indirect access register, which must be programmed before accessing the register DS2 transmit registers.

GN Group Number
 This bit field selects the DS2 group, which can be accessed via the DS2 transmit registers.
 0..6 Group Number.

D2TCFG
DS2 Transmit Configuration Register

Access : read/write
Address : 204_H (PCI), 82_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|----------|----|---|---|
| 15 | | | | | | | | | | | | | | | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LPC(1:0) | E1 | | |

LPC Loopback Code
This bit selects the C-bit which will be inverted when loopback requests are transmitted.
00 Invert 1st C-bit.
01 Invert 2nd C-bit.
10 Invert 3rd C-bit.

E1 G.747 Select
This bit selects the operation mode of the low speed multiplexer.
0 Select M12 mode (4 DS1 into DS2).
1 Select ITU-T G.747 mode (3 E1 into DS2).

D2TCOM
DS2 Transmit Command Register

Access : read/write
Address : 208_H (PCI), 84_H (Local bus)
Reset Value : 0000_H

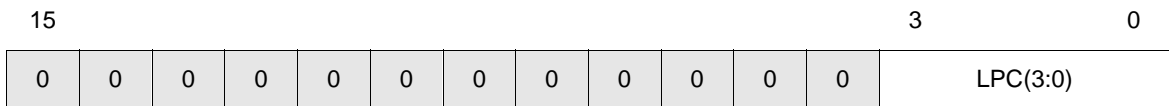
| | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|------------|-----|-----|
| 15 | | | | | | | | | | | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FINSC(1:0) | SRA | RES |

- FINSC** Fault Insertion Code
This bit enables transmission of faults for testing purposes.
- 0 No fault insertion.
 - 1 Insert F-bit errors at low rate (2 out of 5 F-bits).
 - 2 Insert F-bit errors at high rate (2 out of 4 F-bits).
 - 3 Insert M-bit framing bit error (DS1 mode) or P-bit error (ITU-T G.747)
- SRA** Set Remote Alarm
This bit enables transmission of the DS3 remote alarm. In DS1 modes remote alarm is transmitted in subframe 4, block 1 overhead bit and in ITU-T G.747 remote alarm is transmitted in bit 2 of “set II”.
- 0 Normal operation.
 - 1 Enable transmission of remote alarm.
- RES** ITU-T G.747 Reserved Bit
This bit sets the value to be transmitted in the reserved bit of ITU-T G.747 format.
- 0 Transmit reserved bit as '0'.
 - 1 Transmit reserved bit as '1'.

D2TILPC

DS2 Transmit E1/T1 Remote Loopback/Loopback Code Insertion Register

Access : read/write
 Address : 20C_H (PCI), 86_H (Local bus)
 Reset Value : 0000_H



LPC Send Loopback Code for Tributary N
 Setting LPC(x) enables transmission of the loopback code in tributary x.
 The loopback code inserted is specified in D2TCFG.LPC.

| | |
|---|--|
| 0 | Disable transmission of loopback code. |
| 1 | Enable transmission of loopback code. |

D2RSEL
DS2 Receive Group Select Register

Access : read/write
Address : 220_H (PCI), 90_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---------|---|--|
| 15 | | | | | | | | | | | | 2 | 1 | 0 | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | GN(2:0) | | |

Note: This register is an indirect access register, which must be programmed before accessing the register DS2 transmit registers.

GN Group Number
This bit field selects the DS2 group number, which can be accessed via the DS2 receive registers.
0..6 Group Number.

D2RCFG
DS2 Receive Configuration Register

Access : read/write
Address : 224_H (PCI), 92_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|-----|---|-----|-----|---|
| 15 | | | | | | | | | | | | | | 3 | | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ECM | 0 | MFM | FFM | |

Note: ITU-T G.747 mapping and loopback codes are controlled by bits E1 and LPC in the DS3 transmit configuration register D2TCFG.

E1/T1 and loopback codes are controlled by E1 and LPC fields of the D2TCFG register.

- ECM** Error Counter Mode
- DS2 errors are counted in background and copied to foreground (error counter registers) when condition selected via ECM is met.
- 0 Counter values are copied to foreground when copy command is executed. See also register DS3COM.
 - 1 The counter values are copied to the foreground register in one second intervals. At the same time the background registers are reset to zero. This operation is synchronous with the periodic one second interrupt which alerts software to read the register.
- MFM** Multiframe Framing Mode
- This bit selects the M-bit error condition which triggers the DS2 framer to start a new frame search. It is valid in DS1 mode only.
- 0 F-frame search started if 3 contiguous multiframe have M-bit errors.
 - 1 Inhibit new F-frame search due to M-bit errors.
- FFM** F-Framing Mode
- This bit selects the F-bit error condition which triggers the DS2 framer to start a new frame search.
- 0 A new frame search is started when 2 out of 4 contiguous F-bits are in error.
 - 1 A new frame search is started when 2 out of 5 contiguous F-bits are in error.

D2RCOM
DS2 Receive Command Register

Access : read/write
Address : 228_H (PCI), 94_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|------------|---|---|------|-----|--|---|--|--|---|---|
| 15 | | | | | | | | | | 6 | | | | | | 4 | | | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | ESIMC(2:0) | 0 | 0 | C2NC | C2C | | | | | | |

ESIMC Error Simulation Code

This bit field enables error simulation. During error simulation the device generates error interrupts and error status messages. Nevertheless the service is not affected.

- 0 Normal operation (no error simulation)
- 1 Simulate 2 receive F-bit errors/multiframe (186 μsec)
- 2 Simulate 2 receive M-bit errors/multiframe (186 μsec) (DS-1 mode)
Receive parity error/multiframe (133 μsec) (ITU-T G.747 mode)
- 3 Simulate remote alarm
- 4 Simulate loss of frame (RED alarm timer)
- 5 Simulate AIS (AIS alarm timer)
- 6 Simulate receive loop command

C2NC Copy DS2 Error Counters

Only valid when D2RCFG.ECM is set to '0'. Values of DS2 background registers are copied to foreground. Background registers are NOT cleared. Command is self clearing and completes before next register access is possible i.e. software can write command and then immediately read the counters without starting a delay timer.

- 0 No operation.
- 1 Copy background counters to foreground.

C2C Copy and Clear DS2 Error Counters

Only valid when D2RCFG.ECM is set to '0'. Values of DS2 background registers are copied to foreground. Background registers are cleared. Command is self clearing and completes before next register access is

Register Description

possible i.e. software can write command and then immediately read the counters without starting a delay timer.

- 0 No operation.
- 1 Copy background counters to foreground. Clear background counters.

D2RIMSK
DS2 Receive Interrupt Mask Register

Access : read/write
Address : 22C_H (PCI), 96_H (Local bus)
Reset Value : 003F_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|------|------|------|-----|-----|-----|
| 15 | | | | | | | | | | | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LPCS | AISS | REDS | RES | RAS | FAS |

This register provides the interrupt mask for DS2 status interrupts and DS2 loopback code interrupts. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see "Layer One Interrupts" on page 136.

The following definition applies:

- 1 The corresponding interrupt vector will not be generated by the device.
- 0 The corresponding interrupt vector will be generated.

- LPCS Mask 'Loopback Code Status' (flagged in D2RLPCS)
- AISS Mask 'AIS State'
- REDS Mask 'Red Alarm State'
- RES Mask 'Reserved Bit'
- RAS Mask 'DS2 Remote Alarm State'
- FAS Mask 'DS2 Frame Alignment State'

D2RSTAT
DS2 Receive Status Register

Access : read
Address : 230_H (PCI), 98_H (Local bus)
Reset Value : 0001_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|------|------|-----|-----|------|-----|
| 15 | | | | | | | | | | | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AISS | REDS | RES | RAS | COFA | FAS |

Each bit in the DS2 framer receive status register declares a specific condition dependent on the selected modes. The following convention applies to the individual bits:

- 0 The named status is not or no longer existing.
- 1 The named status is currently effective.

The change of status bit can also be used to generate a DS2 interrupt vector. See also register D2RIMSK which describes how to enable/disable interrupt vector generation and refer to the description of DS2 framer interrupts on page "Layer One Interrupts" on page 136.

AISS DS2 AIS Alarm State (unframed all '1's pattern)
AIS is considered valid in a multiframe when fewer than 5 zeros are detected. At 10⁻³ error rates, 1 zero per multiframe is typical. A valid DS2 signal without any bit errors has at least 5 zeros.

The AIS flag nominally changes when the AIS condition is persistent as per alarm timing parameters defined in register D2RAP. The exact time necessary to change the flag could be greater in extremely high error rates. The AIS state is integrated by incrementing or decrementing a counter at the end of each multiframe depending on the AIS condition being valid or invalid respectively.

REDS DS2 Red Alarm State (loss of frame alignment).
The red alarm flag nominally changes when loss of frame alignment condition is persistent as per alarm timing parameters defined in register D2RAP. The exact time necessary to change the flag could be greater if the FAS flag is not constant because the frame alignment state is integrated by incrementing or decrementing a counter at the end of each

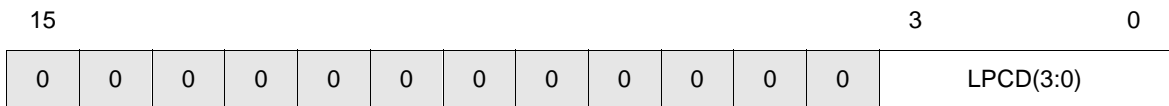
Register Description

multiframe when the FAS flag set or cleared respectively. Note that the framer's verification algorithm is designed to prevent a bouncing FAS flag.

| | |
|------|--|
| RES | Reserved Bit This bit indicates the status of bit 3 in set II of ITU-T G.747 mode. Is it updated if the state persists for at least 8 multiframe. Reserved Bit changes are not reported when the DS2 framer is not aligned. |
| RRA | Remote Alarm This bit indicates that remote alarm is active. Changes are reported when they persist for at least 8 multiframe. In DS1 mode changes on M_x bit are reported, in ITU-T G.747 mode changes of bit 1 of set II are reported. Changes are not reported when the DS2 framer is not aligned. |
| COFA | Change of Frame Alignment. This bit indicates a change of frame alignment event. It is set when the DS2 framer found a new frame alignment and when the new frame position differs from the expected frame position. |
| FAS | Demultiplexer Loss of Frame Alignment This bit indicates that the DS2 framer is not aligned. |

D2RLPCS
DS2 Receive Loopback Code Status Register

Access : read
Address : 234_H (PCI), 9A_H (Local bus)
Reset Value : 0000_H



LPCD(N) Loopback Command Detected

LPCD(x) indicates that a loopback request was received. A loopback request for tributary x is indicated by inverting one of the 3 C-bits of the xth subframe. The C-bit is determined by D2TCFG.LPC. A command state change must persist for 5 contiguous multiframes before it will be reported.

0 No loopback code being received.
1 Loopback code being received.

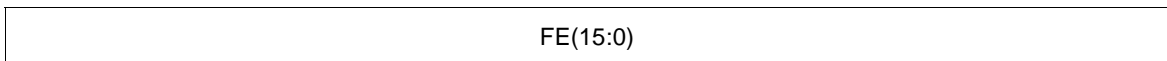
D2RFEC

DS2 Receive Framing Bit Error Counters

Access : read/write
Address : 238_H (PCI), 9C_H (Local bus)
Reset Value : 0000_H

15

0



FE(15:0) Framing Bit Errors
Error counter mode (Clear on Read or Errored Second) depends on register D2RCFG.ECM.
For DS1 mode framing bit errors include F-bit and M-bit errors. For G747 mode, individual bits in the Frame Alignment Signal (FAS) are counted. Errors are not counted in out of frame state.

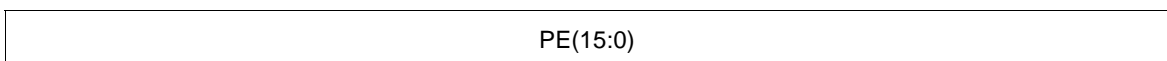
D2RPEC

DS2 Receive Parity Bit Error Counter (ITU-T G.747)

Access : read/write
Address : 23C_H (PCI), 9E_H (Local bus)
Reset Value : 0000_H

15

0



PE(15:0) Parity Errors in ITU-T G.747 mode
Error counter mode (Clear on Read or Errored Second) depends on register D2RCFG.ECM. Errors are not counted in out of frame state.

D2RAP
DS2 Receive Alarm Timer Parameters

Access : read/write
Address : 240_H (PCI), A0_H (Local bus)
Reset Value : 00_H

| | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|-----|----|---------|--|--|--|--|--|--|---|
| 15 | | | | | | | | | 7 | 6 | 5 | | | | | | | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | AIS | CM | CV(5:0) | | | | | | | |

AIS AIS criteria
This bits sets the error rate for AIS detection. Declaration of AIS is specified by bits CM and CV.

ITU-T G.747:

- 0 AIS condition is recognized when the alarm indication signal is received with less than 5 errors in each of 2 consecutive multiframes.
- 1 AIS condition is recognized when the alarm indication signal is received with less than 9 errors in each of 2 consecutive multiframes.

M12 format:

- 0 AIS condition is recognized when the alarm indication signal is received with less than 3 errors in 3156 bits.
- 1 AIS condition is recognized when the alarm indication signal is received with less than 9 errors in 3156 bits.

CM Counter Mode
This bit selects the alarm timer mode. If counter mode is set to multiframes ('0') the value in CV determines the number of multiframes after which the MUNICH256FM declares AIS or RED. When counter mode is set to '½ milliseconds' ('1') the value in CV determines the time in CV x 0.5 ms after which AIS or RED is declared.

- 0 Multiframes.
- 1 ½ Milliseconds.

Register Description

CV Counter Value

Dependent on bit CM the counter value specifies the number of frames or the time in multiples of 0.5 milliseconds when AIS or RED is declared, i.e. setting CV to 20 and CM to '1' sets the alarm integration time to 10 milliseconds.

0..63 Counter Value.

8.2.4 Test Unit Registers

TUTCFG

Test Unit Transmit Configuration Register

Access : read/write
Address : 280_H (PCI), C0_H (Local bus)
Reset Value : 0000_H

| | | | | | | | |
|----|----|-----|----------|---|----------|----|----|
| 15 | 13 | 12 | 8 | 6 | 2 | 1 | 0 |
| 0 | 0 | INV | FBT(4:0) | 0 | LEN(4:0) | ZS | MD |

| | |
|-----|--|
| INV | <p>Invert output</p> <p>This bit enables inversion of the test unit output. Bit inversion is done after the zero suppression insertion point.</p> <p>0 No inversion</p> <p>1 Invert pattern generator output</p> |
| FBT | <p>Feedback Tap</p> <p>This bit field sets the feedback tap in pseudorandom pattern mode. PRBS shift register input bit 0 is XOR of shift register bits LEN and FBT.</p> |
| LEN | <p>Pattern Generator Length</p> <p>This bit field sets the pattern generator length to 1..32.</p> |
| ZS | <p>Enable Zero Suppression</p> <p>This bit enables zero suppression where a '1' bit is inserted at the output if the next 14 bits in the shift register are '0'.</p> <p>0 No zero suppression</p> <p>1 Zero suppression.</p> |
| MD | <p>Generator Mode</p> <p>This bit selects the generator mode of the test unit to be either PRBS or fixed pattern mode.</p> <p>0 Pseudorandom Pattern (PRBS)</p> <p>1 Fixed Pattern</p> |

TUTCOM
Test Unit Transmit Command Register

Access : write
Address : 284_H (PCI), C2_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|------|------|------|------|
| 15 | | | | | | | | | | | | | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | LDER | IN1E | STOP | STRT |

Note: All commands are self clearing i.e. user does not have to clear command. The maximum command rate is limited by clock rate of unit under test and the associated synchronization process. Write interval should be > 4 transmit clock periods e.g. 2.6 μs for T1 tributary test or 634 ns for T2 tributary test.

- LDER** Load Error Rate Register
This bit loads the value of the error rate register TUTEIR to the test unit transmitter. The command can be given while the transmitter is running.
0 No function.
1 Copy value of register TUTEIR to transmit clock region.
- IN1E** Insert One Error in Output
This bit enables a single error insertion in the next bit after command was written.
0 No function
1 Single error insertion.
- STOP** Stop Pattern Generation.
This bit stops the test unit transmitter. When stopped output becomes all '1'.
0 No function.
1 Stop pattern generation.

Register Description

STRT Load/Start Transmitter.

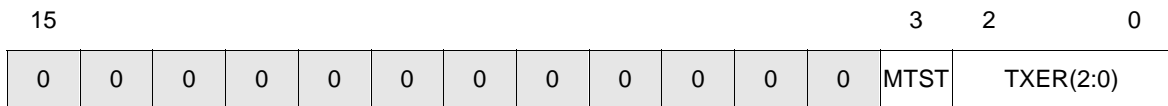
This bit starts the test unit transmitter with the parameters defined in register TUTCFG. In fixed pattern mode the pattern needs to be programmed via register TUTFP0/1 prior to starting the transmitter.

0 No operation.

1 Load/Start test unit.

TUTEIR
Test Unit Transmit Error Insertion Rate Register

Access : read/write
Address : 288_H (PCI), C4_H (Local bus)
Reset Value : 0000_H



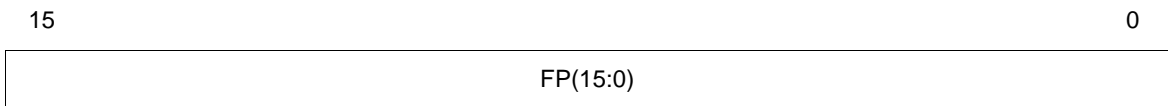
MTST Manufacturing test.
Must be written to '0' for normal operation.

TXER Transmit Error Insertion Rate.
This bit field determines the error insertion rate of the test unit transmitter.

| | |
|-----|------------------------------------|
| 000 | No errors |
| 001 | 10 ⁻¹ (1 in 10) |
| 010 | 10 ⁻² (1 in 100) |
| 011 | 10 ⁻³ (1 in 1 000) |
| 100 | 10 ⁻⁴ (1 in 10 000) |
| 101 | 10 ⁻⁵ (1 in 100 000) |
| 110 | 10 ⁻⁶ (1 in 1 000 000) |
| 111 | 10 ⁻⁷ (1 in 10 000 000) |

TUTFP0
Test Unit Transmit Fixed Pattern Low Word

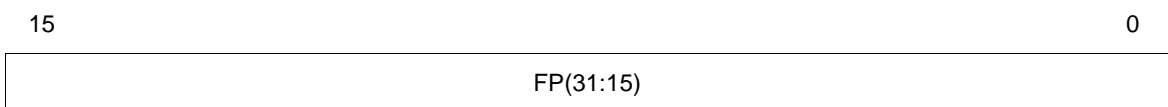
Access : read/write
Address : 28C_H (PCI), C6_H (Local bus)
Reset Value : 0000_H



FP Fixed Pattern Low Word
See description below.

TUTFP1
Test Unit Transmit Fixed Pattern High Word

Access : read/write
Address : 290_H (PCI), C8_H (Local bus)
Reset Value : 0000_H



FP Fixed pattern High Word
The 32 bit fixed pattern is distributed over two 16 bit registers and contains the pattern which is transmitted repetitively from bit FP(TUTCFG.LEN) down to FP(0) when test unit is operated in fixed pattern generator mode.

TURCFG Test Unit Receive Configuration Register

Access : read/write
Address : 2A0_H (PCI), D0_H (Local bus)
Reset Value : 0000_H

| | | | | | | | |
|-----|----|-----|----------|---|----------|----|----|
| 15 | 13 | 12 | 8 | 6 | 2 | 1 | 0 |
| AIM | 0 | DAS | FBT(4:0) | 0 | LEN(4:0) | ZS | MD |

- AIM** Auxiliary Interrupt Mode
This bit field enables the auxiliary interrupt mask AIM of register TURIMSK. In normal operation and if not masked every status event generates an interrupt event. In auxiliary interrupt mode an individual status event generates one interrupt event and further status events of the same class, i.e. 'Bit Error Detected', are masked via an internal mask. This prevents excessive interrupt floods. See register TURIMSK for further details.
0 Normal Operation
1 Auxiliary Interrupt Mode
- DAS** Disable Automatic Synchronization
This bit disables automatic resynchronization in case of high bit error rates. If automatic resynchronization is enabled the receiver automatically tries to resynchronize to the received test pattern.
0 Enable automatic resynchronization.
1 Disable automatic resynchronization.
- FBT** Feedback Tap
This bit field sets the feedback tap of the test unit synchronizer (receiver) in pseudorandom pattern mode. Next input to PRBS reference shift register (bit 0) is XOR of shift register bits LEN and FBT.
- LEN** Reference shift register length
This bit field sets the length of the receiver's test pattern register.

Register Description

| | |
|----|--|
| ZS | <p>Enable Zero Suppression</p> <p>This bit enables zero suppression at the test unit receiver. A '1' is expected and inserted at the input if the next 14 bits in the shift register are set to '0'.</p> <p>0 No zero suppression.</p> <p>1 Enable zero suppression.</p> |
| MD | <p>Generator Mode</p> <p>This bit sets the generator mode of the test unit to either PRBS or fixed pattern.</p> <p>0 Pseudorandom Pattern (PRBS)</p> <p>1 Fixed Pattern</p> |

TURCOM
Test Unit Receive Command Register

Access : write
Address : 2A4_H (PCI), D2_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|-----|-----|------|------|------|
| 15 | | | | | | | | | | | | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RDF | RDC | CAIM | STOP | STRT |

Note: All commands are self clearing i.e. user does not have to clear command. The maximum command rate is limited by clock rate of unit under test and the associated synchronization process. Write interval should be > 4 transmit clock periods e.g. 2.6 μs for T1 tributary test or 634 ns for DS2 tributary test.

- RDF** Copy Receiver's 32 bit Pattern
This bit loads the test units internal receiver pattern to register TURFP in fixed pattern mode. In synchrones state TURFP will be loaded with the pattern received. In asynchronous state TURFP with a 32-bit sample of the last received bit stream.
- 0 No function.
1 Update register TURFP with synchronizer pattern.
- RDC** Copy bit counter and error counter
This bit loads the test units internal bit counter and error counter to registers TURBC0,1 and TUREC0,1.
- 0 No function.
1 Copy counter.
- CAIM** Clear Auxiliary Interrupt Masks.
This bit resets the internal auxililiary mask. See TURCFG.AIM.
- 0 no operation
1 clear auxiliary interrupts
- STRT** Start Receiver.
This bit loads and starts the test unit receiver with the parameters defined in register TURCFG.

Register Description

- 0 No operation.
- 1 Load/Start test unit receiver.

TURERMI
Test Unit Receive Error Measurement Interval Register

Access : read/write
Address : 2A8_H (PCI), D4_H (Local bus)
Reset Value : 0000_H

| | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|-----|-----------|
| 15 | | | | | | | | | | | 3 | 2 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | TST | RXMI(2:0) |

TST Test Mode
This bit enables measurement interval timer test.

0 Normal operation

1 Auto test of measurement interval function. End of Measurement interrupt should be asserted after approximately 4250 receive clock cycles (if enabled). The lower three bits of register FPAT should be “111”.

RXMI Receive Error Rate Measurement Interval
This bit field defines the measurement interval in terms of input bits for measurement of receive bit error rate.

At the end of the measurement window, contents of background error counter are automatically copied to foreground error counter and reset for next measurement interval. An interrupt can be generated at the end of each measurement interval.

000_B Max measurement interval of 2³²-1

001_B 10³ bits

010_B 10⁴ bits

011_B 10⁵ bits

100_B 10⁶ bits

101_B 10⁷ bits

110_B 10⁸ bits

111_B 10⁹ bits

TURIMSK
Test Unit Receive Interrupt Mask Register

Access : read/write
Address : 2AC_H (PCI), D6_H (Local bus)
Reset Value : 001F_H

| | | | | | | | | | | | |
|----|---|----|----------|---|---|---|------|-----|------|-----|-----|
| 15 | | 12 | | 8 | | 4 | 3 | 2 | 1 | 0 | |
| 0 | 0 | 0 | AIM(4:0) | 0 | 0 | 0 | ERXM | BED | ALL1 | LOS | SYN |

This register provides the interrupt masks for the test unit interrupts. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see "Layer One Interrupts" on page 136.

The following definition applies:

- 1 The corresponding interrupt vector will not be generated by the device.
- 0 The corresponding interrupt vector will be generated.

- ERXM Mask 'End of Receive Error Rate Measurement'
- BED Mask 'Bit Error Detected'
- ALL1 Mask 'All '1' Pattern Received'
- LOS Mask 'Loss of Signal'
- SYN Mask 'Change in Receiver Synchronization State'

AIM flags have same layout as the above five mask but are internal masks that are set automatically following the interrupt in the AIM mode. This mask prevents excessive bus load in error conditions. AIM flags are cleared by the **TURCOM.CAIM** command. They are "read only" flags in this register.

TURSTAT Test Unit Receive Status Register

Access : read
Address : 2B0_H (PCI), D8_H (Local bus)
Reset Value : 0021_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|------|-----|-----|------|-----|-----|----|----|-----|
| 15 | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | INVS | LA1 | LA0 | LOOS | EMI | LBE | A1 | A0 | OOS |

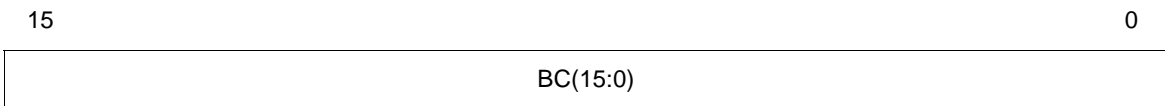
- INV** Inverted Pattern
This bit indicates that the received PRBS sequence is inverted.
0 Not Inverted.
1 Inverted.
- LA1** Latched 'Input all '1"
This bit indicates that the condition 'Input all '1" was active since last status register read.
- LA0** Latched 'Input all '0"
This bit indicates that the condition 'Input all '0" was active since last status register read.
- LOOS** Latched Out of Synchronization
This bit indicates that the receiver was out of synchronization since last status register read.
- EMI** End of Measurement Interval
This bit indicates that the end of the measurement interval was reached since last read of error counter or that command TURCMD.RDC was given. The results of the bit error rate test are available in register TURBC0,1 and TUREC0,1. This flag is cleared when the error counter is read. Counters will not be overwritten while EMI is '1'.
- LBE** Latched Bit Error Detected Flag
This bit indicates that at least '1' one bit error occurred since last read of this register. It is cleared by status register read.
- A1** Input all '1's
This bit indicates that the input contained all '1' during the last 32 bits. It is reset if at least one '0' occurs in 32 bits.

Register Description

| | |
|-----|---|
| A0 | Input all '0's This bit indicates that the input contained all '0' during the last 32 bits. It is reset if at least one '1' occurs in 32 bits. |
| OOS | Receiver Out of Synchronization This bit indicates the status of the test unit synchronizer. |

TURBC0
Test Unit Receive Bit Counter Low Word

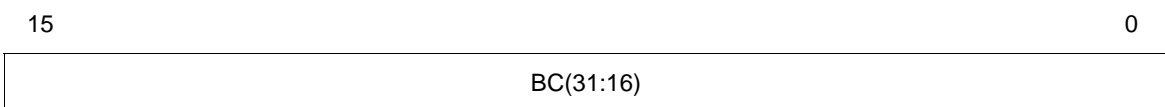
Access : read
Address : 2B4_H (PCI), DA_H (Local bus)
Reset Value : 0000_H



BC(31:0) Bit Counter
See description below.

TURBC1
Test Unit Receive Bit Counter High Word

Access : read
Address : 2B8_H (PCI), DC_H (Local bus)
Reset Value : 0000_H



BC(31:0) Bit Counter

BC is a 32 bit counter which is split between two 16 bits registers. It counts receive clock slots when the receiver is enabled. Bits are counted in a background register which is not directly readable. The values are transferred to the two 16 bit foreground (readable) registers and cleared in one of the two ways:

1. Assert command TURCOM.RDC.
2. Automatically at end of measurement interval.

The background register is transferred to the foreground register and cleared in the same way as the bit error counter (see previous section).

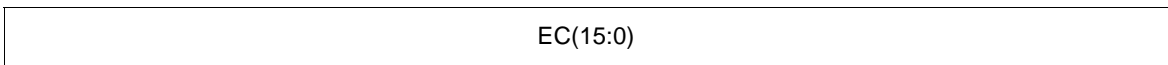
Register Description

When the error registers are read in response to the “End of Measurement Interval” interrupt vector , reading this register is not necessary because the measurement interval would be known. However the user could assert command TURCOM.RDC to terminate the measurement interval early and transfer the current bit error count and bit count to the foreground registers (polling mode).

TUREC0
Test Unit Receive Error Counter Low Word

Access : read
Address : 2BC_H (PCI), DE_H (Local bus)
Reset Value : 0000_H

15 0

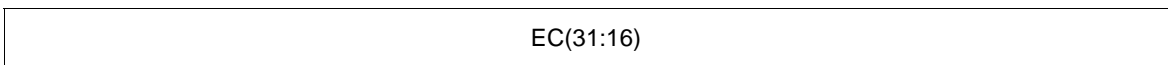


EC(31:0) Error Counter
See description below.

TUREC1
Test Unit Receive Error Counter High Word

Access : read
Address : 2C0_H (PCI), E0_H (Local bus)
Reset Value : 0000_H

15 0



EC(31:0) Error Counter

This 32 bit counter counts receive errors detected when receiver is enabled and in synchronized state. When the 'Bit Error Detected' interrupt is enabled, it will be asserted and then automatically masked when this counter is incremented.

Errors are counted in a background register (not directly readable) until:

1. The user asserts command TURCOM.RDC.
2. The end of measurement interval is reached and the last result was read.

In both cases the value of the background register is copied to TUREC.EC and the measured values are accessible. An 'End of

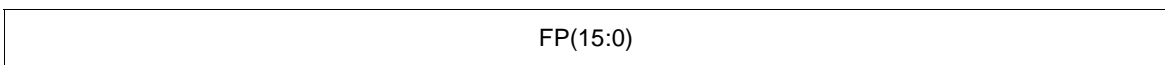
Register Description

Receive Error Rate Measurement' interrupt vector is optionally generated.

TURFP0
Test Unit Receive Fixed Pattern Low Word

Access : read
Address : 2C4_H (PCI), E2_H (Local bus)
Reset Value : 0000_H

15 0

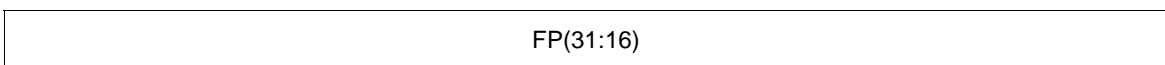


FP(31:0) Fixed pattern
See description below.

TURFP1
Test Unit Receive Fixed Pattern High Word

Access : read
Address : 2C8_H (PCI), E4_H (Local bus)
Reset Value : 0000_H

15 0



FP(31:0) Fixed Pattern

This 32 bit field is distributed over two 16 bit registers and is used in the fixed pattern mode (TURCFG.MD='1'). The TURCOM.RDF command will copy the current state of the receiver's 32 bit pattern generator to this register. If the receiver is synchronized, bits FP(TURCFG.LEN:0) contain the fixed pattern being received. Bit 0 is the most recently received. If not synchronized, the register contains a 32 bit sample of input data.

8.2.5 Transmit Framer Register

TCMDR T1/E1 Transmit Command Register

Access : read/write

Address : 00_H

Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|-----|-------|------|-----|-----|-----|
| 15 | | | | | | | | | | | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | XAP | XPRBS | XAIS | XRA | XLU | XLD |

XAP Transmit Auxiliary Pattern

This bit enables transmission of auxiliary pattern in the outgoing bit stream. The auxiliary pattern is defined as a continuous pattern of '01'.

0 Disable transmission of auxiliary pattern.

1 Enable transmission of auxiliary pattern. This function is not available if bit XAIS is set to '1'.

XPRBS Transmit PRBS

This bit enables the transmission of the pseudo-random bit sequence defined in register TPRBSC.

0 Disable transmission of PRBS.

1 Enable transmission of PRBS.

XAIS Transmit AIS

This bit enables transmission of alarm indication signal towards the remote end. AIS is an all one unframed signal.

0 Disable transmission of AIS.

1 Enable transmission of AIS.

Register Description

| | |
|-----|--|
| XRA | Transmit Remote Alarm (Yellow Alarm) This bit enables the transmission of remote alarm in the outgoing bit stream. Clearing the bit will remove the remote alarm pattern. T1 0 Disable transmission of remote alarm. 1 Enable transmission of remote alarm. Remote alarm pattern is selected via register FMR.SRAF. E1 0 Disable transmission of remote alarm. 1 Set A-bit in transmitted service word. |
| XLU | Transmit Line Loopback Actuate (Up) Code 0 Normal operation. 1 A one in this bit position will cause the transmitter to replace normal transmit data with the line loopback actuate code continuously until this bit is reset. The line loopback actuate code will be optionally overwritten by the framing/DL/CRC bits. |
| XLD | Transmit Line Loopback Deactuate (Down) Code 0 Normal operation. 1 A one in this bit position will cause the transmitter to replace normal transmit data with the line loopback deactuate code continuously until this bit is reset. The line loopback deactuate code will be optionally overwritten by the framing/DL/CRC bits. |

TFMR
T1/E1 Transmit Mode Register

Access : read/write
Address : 01_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|-----|------|------|------|---------|---|
| 15 | | | | | | | | | | | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | XAS | AXRA | SRAF | T1E1 | FM(1:0) | |

XAS Automatic Spare Bit Insertion

E1: CRC-4 Multiframe

- 0 Normal operation. Content of register XSP.XS13 and XSP.XS15 is inserted in the E-Bit of time slot 0 in frame 13 and frame 15 respectively.
- 1 Submultiframe status will be automatically set in the outgoing data stream. Each received, errored submultiframe causes bit one of time slot 0 of frame 13 and frame 15 to be '0'. Otherwise these bits are set to '1'.

AXRA Automatic Transmit Remote Alarm

Setting this bit enables automatic transmission of remote alarm.

- 0 Normal operation.
- 1 The Remote Alarm (yellow alarm) bit will be automatically set in the outgoing data stream if the receiver is in asynchronous state (FRS.LFA bit is set). In synchronous state the remote alarm bit will be reset.

Register Description

| | |
|------|---|
| SRAF | <p>Select Remote (Yellow) Alarm Format</p> <p>Setting this bit enables the remote alarm format in T1 mode. This bit has no function in E1 mode.</p> <p>T1: F4</p> <p>1 Bit 2 = 0 in every channel</p> <p>T1: F12</p> <p>0 FS bit of frame 12.</p> <p>1 Bit 2 = 0 in every channel.</p> <p>T1: ESF</p> <p>0 Pattern '1111 1111 0000 0000...' in data link channel.</p> <p>1 Bit 2 = 0 in every channel.</p> |
| T1E1 | <p>T1/E1 mode selection</p> <p>This bit switches the transmit framer into T1 and E1 mode.</p> <p>0 Select T1 mode.</p> <p>1 Select E1 mode.</p> |
| FM | <p>Select Frame Mode</p> <p>This bit field determines the framing mode of the transmit framer.</p> <p>T1</p> <p>00_B Select ESF format.</p> <p>01_B Select F12 format.</p> <p>10_B Select F4 format.</p> <p>Other Reserved</p> <p>E1</p> <p>00_B Select Double frame format.</p> <p>01_B Select CRC-4 multiframe format.</p> <p>Other Reserved</p> |

TLCR0
T1/E1 Transmit Loop Code Register 0

Access : read/write
Address : 02_H
Reset Value : 0000_H

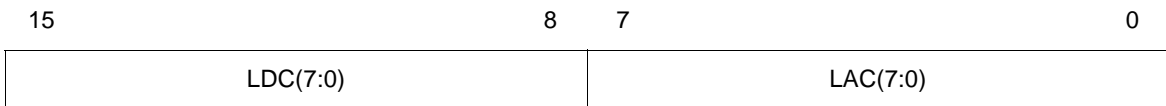
| | | | | | | | | | | | | | | |
|------|-----|---|---|---|---|-----------|---|---|---|---|---|---|-----------|---|
| 15 | 14 | | | | | 9 | 8 | | | | | | 1 | 0 |
| FLLB | LCS | 0 | 0 | 0 | 0 | LDCL(1:0) | 0 | 0 | 0 | 0 | 0 | 0 | LACL(1:0) | |

- FLLB** Disable Framed Line Loopback
This bit switches between framed and unframed transmission of line loopback. In unframed transmission the FS/DL bit the line loopback code overwrites the FS/DL bits, while in framed transmission the FS/DL bits will not be overwritten by the line loopback code.
0 Set framed line loopback transmission.
1 Set unframed line loopback transmission.
- LCS** Loop Code Select
This bit switches between line loopback code defined in ANSI T1.403 or a user definable loopback code defined in register TLCR1.
0 Select ANSI codes.
1 Select line loopback code defined in register TLCR1.
- LDCL** Line Loopback Deactuate Code Length
This bit field determines the length of the line loopback deactuate code specified in register TLCR1. The length of the loopback code can be specified in a range of 5 to 8 bits.
00_B..11_B Specifies code length in the range of 5 to 8 bits.
- LACL** Line Loopback Actuate Code Length (5-8 bit)
This bit field determines the length of the line loopback actuate code specified in register TLCR1. The length of the loopback code can be specified in a range of 5 to 8 bits.
00_B..11_B Specifies code length in the range of 5 to 8 bits.

Note: Codes of smaller length might be activated by multiple entry, e.g. code 001: write 001001 to TLCR1 register and define code length of 6 bits.

TLCR1
T1/E1 Transmit Loop Code Register 1

Access : read/write
Address : 03_H
Reset Value : 0000_H



LDC Line Loopback Deactuate Code
This bit field is sent in the outgoing bit stream if enabled via bit TCMDR.XLD and TLCR0.LCS.

Note: Most significant bit is sent first. E.g. TLCR0.LDCL = 01_B specifies code length to be six bits long. In this case LDC(5) is sent first.

LAC Line Loopback Actuate Code
This bit field is sent in the outgoing bit stream if enabled via bit TCMDR.XLU and TLCR0.LCS.

Note: Most significant bit is sent first. E.g. TLCR0.LACL = 01_B specifies code length to be six bits long. In this case LAC(5) is sent first.

TPRBSC T1/E1 Transmit PRBS Control Register

Access : read/write

Address : 04_H

Reset Value : 001F_H

| | | | | | | | | | | |
|-------|---|----|-------|---|---|----------|---|---|---|----------|
| 15 | | 12 | | 9 | 8 | | 4 | 0 | | |
| FPRBS | 0 | 0 | IPRBS | 0 | 0 | PRP(1:0) | 0 | 0 | 0 | FPL(4:0) |

FPRBS Framed PRBS

This bit field enables framed or unframed transmission of the pseudo-random bit sequence.

0 Transmit framed PRBS.

1 Transmit unframed PRBS.

IPRBS Invert PRBS

This bit field enables inversion of the pseudo-random bit sequence in transmit direction.

0 PRBS is not inverted.

1 PRBS is inverted.

PRP Pseudo-Random Pattern

This bit field determines the generator polynomial for the pseudo-random bit sequence.

00_B PRBS is generated according to $2^{15} - 1$ (ITU-T O. 151)

01_B PRBS is generated according to $2^{20} - 1$ (ITU-T O. 151)

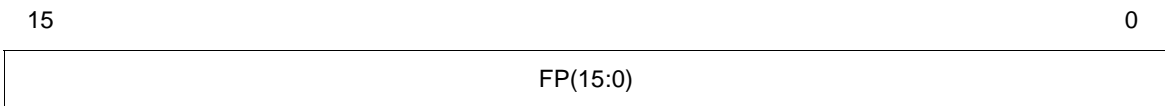
1-_B For PRBS the fixed pattern, defined in TFPR0 and TFPR1, is selected.

FPL Fixed Pattern Length

This bit field sets the length of the fixed pattern FP which is located in register TFPR0 and TFPR1. E.g.: FPL(4:0) = 10010_B means pattern length is equal to 19, which implies that the bits FP(18)..FP(0) form the PRBS.

TFPR0
T1/E1 Transmit Fixed Pattern Register Low Word

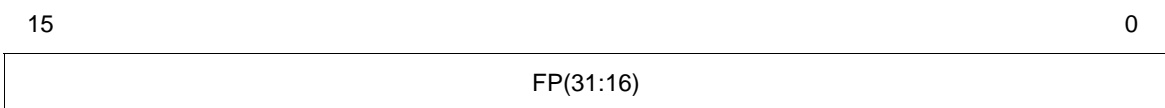
Access : read/write
Address : 05_H
Reset Value : 0000_H



FP(31:0) Fixed Pattern Low Bytes
See description below.

TFPR1
T1/E1 Transmit Framer Fixed Pattern Register High Word

Access : read/write
Address : 06_H
Reset Value : 0000_H



FP(31:0) Fixed Pattern High Bytes
This bit field together with bit field TFPR0.FP defines a bit sequence, which can be sent instead of a pseudo-random bit sequence. FP is sent in the order FP(TPRBSC.FPL-1) down to FP(0) and will be repeated until deactivated.

TPTSL0

T1/E1 Transmit PRBS Time Slot Number Register Low Word

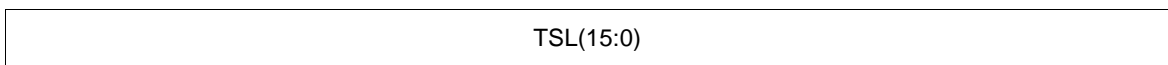
Access : read/write

Address : 07_H

Reset Value : FFFF_H

15

0



TSL(31:0) Time slot 15..0 Select
See description below.

TPTSL1

T1/E1 Transmit PRBS Time Slot Number Register High Word

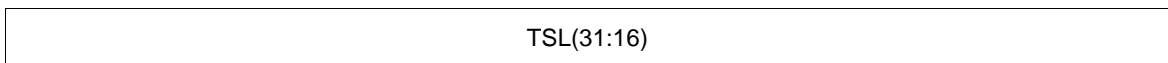
Access : read/write

Address : 08_H

Reset Value : 00FF_H

15

0



TSL(31:0) Time slot 31..16 Select

Selected bits in bit field TSL and TPTSL0.TSL determine those time slots, which are used for PRBS generation. Time slots can be programmed arbitrarily. E.g. if TPTSL0.TSL(1) and TPTSL0.TSL(2) are set to '1', the PRBS is sent continuously over both time slots combined.

XSP
T1/E1 Transmit Spare Bit Register

Access : read/write
Address : 09_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------|------|
| 15 | | | | | | | | | | | | | | | | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | XS13 | XS15 |

XS13, XS15 Transmit Spare Bit

E1: CRC-4 Multiframe

Dependent on bit FMR.XAS and framer mode spare bits of service word in CRC-4 multiframe 13 and 15 are replaced by XS13 and XS15.

8.2.6 Receive Framer Registers

RCMDR T1/E1 Receive Command Register

Access : read/write
Address : 00_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|----------|---|--|-----|---|---|
| 15 | | | | | | | | | | | 5 | 4 | | | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SIM(3:0) | | | FRS | | |

SIM Alarm Simulation

This bit field enables alarm simulation in the receive framer. See codes for specific function.

0000_B Disable alarm simulation.

0001_B Simulate loss of signal

Setting this code:

- Generate 'Loss of Signal Status' interrupt vector.
- Flag 'Loss of Signal' via bit FSR.LOS.
- Generate PDEN interrupt vector.
- Flag 'Pulse Density Code Violation Detected' via bit FSR.PDEN/AUX.

Removing this code:

- Generate 'Loss of Signal Status' interrupt vector.
- Remove signalling of 'Loss of Signal'.
- Generate PDEN interrupt vector.
- Remove signalling of 'Pulse Density Code Violation Detected'.

0010_B Simulate Alarm Indication Signal

Setting this code:

- Generate 'Loss of Frame Alignment' interrupt vector.
- Flag 'Loss of Frame Alignment' via bit FRS.LFA.
- Generate 'Alarm Indication Signalled' interrupt vector.
- Flag 'Alarm Indication Signalled' via bit FRS.AIS.

Register Description

Removing this code:

- Generate 'Loss of Frame Alignment Status' interrupt vector.
- Remove signalling of 'Loss of Frame Alignment'.
- Generate 'Alarm Indication Signal Status' interrupt vector.
- Remove signalling of 'Alarm Indication Signalled'.

0011_B Simulate auxiliary pattern ('...010101...' pattern)

This sequence simulates also loss of frame (required for auxiliary pattern).

Setting this code:

- Generate 'Auxiliary Pattern Status' interrupt vector.
- Generate 'Loss of Frame Alignment Status' interrupt vector.
- Flag 'Loss of Signal' via bit FRS.LFA.
- Flag 'Auxiliary Pattern detected' via bit FRS.PDEN/AUX.
- Flag 'Loss of Multiframe Alignment' via bit FRS.LMFA (CRC-4 Multiframe mode).
- Increment framing error counter by 3 or 4 depending on RFMR.SSP

Removing this code:

- Generate 'Auxiliary pattern Status' interrupt vector.
- Generate 'Loss of Frame Alignment Status' interrupt vector.
- Remove signalling of 'Loss of Frame Alignment'.
- Remove signalling of FRS.PDEN/AUX.
- Remove signalling of 'Loss of Multiframe Alignment'.

0100_B Simulate loss of frame

Setting this code:

- Generate 'Loss of Frame Alignment Status' interrupt vector.
- Flag 'Loss of Signal' via bit FRS.LFA.
- Flag 'Loss of Multiframe Alignment' via bit FRS.LMFA (CRC-4 multiframe mode).
- Increment framing error counter by 2, 3, or 4 (depends on RFMR.SSP).
- Increment errored seconds (T1 mode only).

Removing this code:

- Generate 'Loss of Frame Alignment Status' interrupt vector.
- Remove signalling of 'Loss of Frame Alignment'.
- Remove signalling of 'Loss of Multiframe Alignment'.

Register Description

0101_B Simulate remote alarm

Setting this code:

- Generate 'Remote Alarm Status' interrupt vector.
- Flag 'Received Remote Alarm' bit FRS.RRA.

Removing this code:

- Generate 'Remote Alarm Status' interrupt vector.
- Remove signalling of 'Receive Remote Alarm'.

0110_B Simulate CRC error (T1 ESF or E1 CRC-4 multiframe mode)

Setting this code:

- Generate CRC interrupt vector.
- Increment CRC error counter.

Removing this code:

- Stop generation of CRC interrupt vector.
- Stop increment of CRC error counter.

FRS

Force Resynchronization

A transition from low to high will force the frame aligner to execute a resynchronization of the pulse frame. The procedure depends on the status of bit FMR.SSP.

0 No operation.

1 Change from '0' to '1' forces resynchronization.

RFMR
T1/E1 Receive Mode Register

Access : read/write
Address : 01_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|------|------|------|------|-----|----------|---|------|------|---------|---|---|
| 15 | | | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | LOSR | ALMF | RRAM | AIS3 | SSP | SSC(1:0) | 0 | SRAF | T1E1 | FM(1:0) | | |

LOSR Loss of Signal Recovery

This bit sets the conditions for 'Loss of Signal' detection.

T1

- 0 Loss of signal cleared, when pulse density defined by register PCR is detected during a time interval declared by register PCD.
- 1 Loss of signal cleared, when pulse frame density defined by register PCR is detected during a time interval declared by register PCD and a pulse density of at least N '1's in every N+1 octets (0<N<24) during recovery interval defined in register PCD is detected.

E1

- 0 Loss of signal cleared, when pulse density defined by register PCR is detected during a time interval declared by register PCD.
- 1 No function.

ALMF Automatic Loss of Multiframe

This bit selects condition for automatic loss of multiframe.

T1

- 0 CRC errors do not cause loss of frame alignment.
- 1 320 or more CRC errors in one second cause loss of frame alignment.

E1

- 0 CRC errors do not cause loss of frame alignment.
- 1 915 or more CRC-4 errors in one second cause loss of frame alignment.

RRAM

Receive Remote Alarm Mode

The conditions for remote (yellow) alarm detection can be selected via this bit to allow detection even in the presence of BER 10^{-3} . Remote alarm detection is flagged in register FRS.RRA and can be signalled as an interrupt.

T1: F4

0 Normal operation

Detection:

Bit 2 = '0' in every speech channel per frame.

Release:

The alarm will be reset when above conditions are no longer detected.

1 Detection with BER 10^{-3}

Detection:

Bit 2 = '0' in 255 consecutive speech channels.

Release:

The alarm will be reset when receiver does not detect the Bit 2 = '0' condition for three consecutive pulseframes.

T1: F12

0 Normal operation

Depending on bit FMR0.SRAF:

0 Detection:

FS-bit of frame 12 is forced to '1'.

Release:

The alarm will be reset when above conditions are no longer detected.

1 Detection:

Bit 2 = '0' in every speech channel per frame.

Release:

The alarm will be reset when above conditions are no longer detected.

1 Detection with BER 10^{-3}

Remote alarm detection depending on bit FMR0.SRAF:

0 Detection:

FS-bit of frame 12 is forced to '1'.

Release:

The alarm will be reset when receiver does not detect the 'Fs-bit' condition for three consecutive multiframe.

Register Description

- 1 Detection:
Bit 2 = '0' in 255 consecutive speech channels.

Release:
The alarm will be reset when receiver does not detect the Bit 2 = '0' condition for three consecutive pulseframes.

T1: ESF

- 0 Normal operation
Remote alarm detection depending on bit FMR0.SRAF:
 - 0 Detection
Pattern '1111 1111 0000 0000...' in data link channel.

Release:
The alarm will be reset when above conditions are no longer detected.
 - 1 Detection:
Bit 2 = '0' in every speech channel per frame.

Release:
The alarm will be reset when above conditions are no longer detected.
- 1 Detection with BER 10^{-3}
Remote alarm detection depending on bit FMR0.SRAF:
 - 0 Detection
Pattern '1111 1111 0000 0000...' in data link channel.

Release:
The alarm will be reset when receiver does not detect 'DL pattern' for three times in a row.
 - 1 Detection:
Bit 2 = '0' in 255 consecutive speech channels.

Release:
The alarm will be reset when receiver does not detect the Bit 2 = '0' condition for three consecutive pulseframes.

AIS3

Select AIS Condition

This bit selects the condition which leads to AIS reporting.

T1: F4, F12

- 0 AIS (blue alarm) is indicated, when two or less zeros in the received bit stream are detected in a time interval of 12 frames.
- 1 AIS (blue alarm) detection is only enabled, when framer is in asynchronous state. The alarm is indicated, when three or less

Register Description

zeros within a time interval of 12 frames are detected in the received bit stream.

T1: ESF

- 0 AIS (blue alarm) is indicated, when two or less zeros in the received bit stream are detected in a time interval of 24 frames.
- 1 AIS (blue alarm) detection is only enabled, when framer is in asynchronous state. The alarm is indicated, when five or less zeros within a time interval of 24 frames are detected in the received bit stream.

SSP

Select Synchronization/Resynchronization Procedure

T1: F12

- 0 Specified number of errors in FT framing or specified number of errors in FS framing leads to loss of synchronization (FRS.LFA). In the case of FS bit framing errors, bit FRS.LMFA is set additionally. A complete new synchronization procedure is initiated to regain pulseframe alignment and then multiframe alignment.
- 1 Specified number of errors in FT framing has the same effect as above. Specified number of errors in FS framing only initiates a new search for multiframe alignment without influencing pulseframe synchronous state (FRS.LMFA is set).

T1: ESF

- 0 Synchronization is achieved only on verification of the framing pattern.
- 1 Synchronous state is reached when framing pattern and CRC-6 checksum are correctly found.

SSC

Select Synchronization Conditions

T1

Loss of Frame Alignment (FRS.LFA or opt. FRS.LMFA) is declared if

- 00_B 2 out of 4 framing bits
- 01_B 2 out of 5 framing bits
- 10_B **F12**
2 out of 6 framing bits
- ESF**
2 out of 6 framing bits per multiframe period
- 11_B 4 consecutive incorrect multiframe pattern

It depends on the selected multiframe format and optionally on bit FMR.SSP which framing bits are observed:

Register Description

F12 SSP = 0: FT bits → FRS.LFA: FS bits → FRS.LFA and FRS.LMFA
SSP = 1: FT → FRS.LFA
FS → FRS.LMFA

ESF ESF framing bits → FRS.LFA

E1

00_B 3 out of 4 consecutive FAS or service word errors

01_B 4 out of 4 consecutive FAS or service word errors

10_B 3 out of 3 FAS errors

11_B 4 out of 4 FAS errors

SRAF

Select Remote (Yellow) Alarm Format

This bit is valid for T1 mode only.

T1: F4

0/1 Bit 2 = '0' in every channel.

T1: F12

0 FS bit of frame 12.

1 Bit 2 = '0' in every channel.

T1: ESF

0 Pattern '1111 1111 0000 0000...' in data link channel.

1 Bit 2 = '0' in every channel.

T1E1

T1/E1 Mode Selection

This bit switches the receive framer into T1 or E1 mode.

0 Select T1 mode.

1 Select E1 mode.

FM

Select Frame Mode

This bit field selects the framing mode of the receive framer.

T1

00_B ESF-Format

01_B F12-Format

10_B F4-Format

Other Reserved

E1

00_B Doubleframe

01_B CRC-4

10_B CRC-4 Interworking mode

Other Reserved

RLCR0
T1/E1 Receive Loop Code Register 0

Access : read/write
Address : 02_H
Reset Value : 0000_H

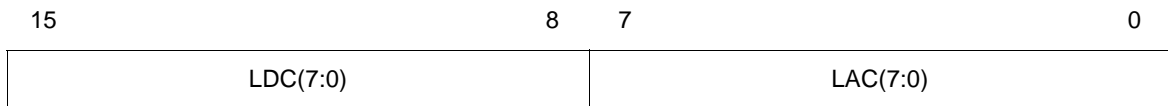
| | | | | | | | | | | | | | | |
|----|-----|---|---|---|---|-----------|---|---|---|---|---|---|-----------|---|
| 15 | 14 | | | | | 9 | 8 | | | | | | 1 | 0 |
| 0 | LCS | 0 | 0 | 0 | 0 | LDCL(1:0) | 0 | 0 | 0 | 0 | 0 | 0 | LACL(1:0) | |

- LCS** Loop Code Select
This bit switches between line loopback code defined in ANSI T1.403 or a user definable loopback code defined in register RLCR1.
0 Select ANSI codes.
1 Select line loopback code defined in register RLCR1.
- LDCL** Line Loopback Deactuate Code Length
This bit field determines the length of the line loopback deactuate code specified in register TLCR1. The length of the loopback code can be specified in a range of 5 to 8 bits.
00_B..11_B Specifies code length in the range of 5 to 8 bits.
- LACL** Line Loopback Actuate Code Length (5-8 bit)
This bit field determines the length of the line loopback actuate code specified in register TLCR1. The length of the loopback code can be specified in a range of 5 to 8 bits.
00_B..11_B Specifies code length in the range of 5 to 8 bits.

Note: Codes of smaller length might be activated by multiple entry, e.g. code 001: write 001001 to LCR1 register and define code length of 6 bits.

RLCR1
T1/E1 Receive Loop Code Register 1

Access : read/write
Address : 03_H
Reset Value : 0000_H



LDC Line Loopback Deactuate Code
This incoming bit stream will be compared against this bit field if enabled via bit RLCR0.LCS.

Note: Most significant bit is sent first. E.g. TCLR0.LDCL = 01_B specifies code length to be six bits long. In this case LDC(5) is sent first.

LAC Line Loopback Actuate Code
This incoming bit stream will be compared against this bit field if enabled via bit RLCR0.LCS.

Note: Most significant bit is sent first. E.g. TCLR0.LACL = 01_B specifies code length to be six bits long. In this case LAC(5) is sent first.

RPRBSC
T1/E1 Receive PRBS Control Register

Access : read/write
Address : 04_H
Reset Value : 001F_H

| | | | | | |
|----|----|------|----------|---|----------|
| 15 | 13 | 9 | 8 | 4 | 0 |
| 0 | 0 | EPRM | 0 | 0 | 0 |
| | | | PRP(1:0) | 0 | 0 |
| | | | | 0 | FPL(4:0) |

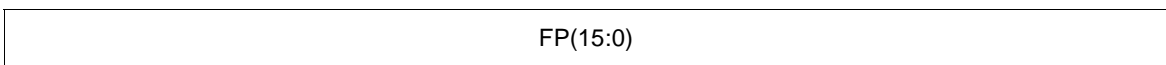
- EPRM** Enable PRBS Monitor
This bit enables the PRBS monitoring function. When PRBS monitor is enabled the pseudo-random pattern synchronizer logs onto the pseudo-random pattern defined in PRB.
0 PRBS monitor is disabled.
1 PRBS monitor is enabled.
- PRP** Pseudo-Random Pattern
00_B The incoming pattern is compared according to $2^{15} - 1$ (ITU-T O.151)
01_B The incoming pattern is compared according to $2^{20} - 1$ (ITU-T O.151)
11_B The incoming pattern is compared to the fixed pattern, defined in RFPR0 and RFPR1. The pattern length is defined in FPL.
Other Reserved
- FPL** Fixed Pattern Length, e.g.: =10010 means pattern length is equal to 19, which implies that the bits RFPR1/0.FP(18)..FP(0) form the PRBS.

RFPR0
T1/E1 Receive Fixed Pattern Register Low Word

Access : read/write
Address : 05_H
Reset Value : 0000_H

15

0



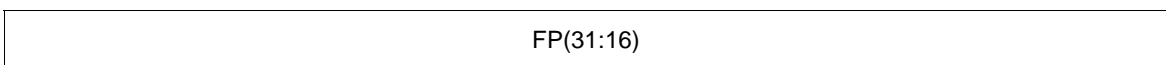
FP Fixed Pattern Low Bytes
See description below.

RFPR1
T1/E1 Receive Fixed Pattern Register High Word

Access : read/write
Address : 06_H
Reset Value : 0000_H

15

0



FP Fixed Pattern High Bytes
This bit field together with RFPR0.FP defines a bit sequence, which will be monitored in the PRBS synchronous state. FP is compared in the order FP(RPRBSC.FPL-1) down to FP(0) and comparison will be repeated until deactivated.

RPTSL0

T1/E1 Receive PRBS Time Slot Number Register Low Word

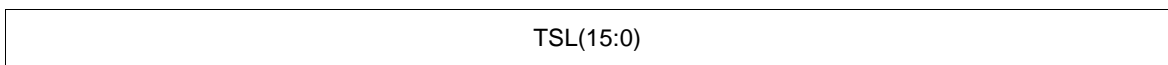
Access : read/write

Address : 07_H

Reset Value : FFFF_H

15

0



TSL Time slot 15..0 Select
See description below.

RPTSL1

T1/E1 Receive PRBS Time Slot Number Register High Word

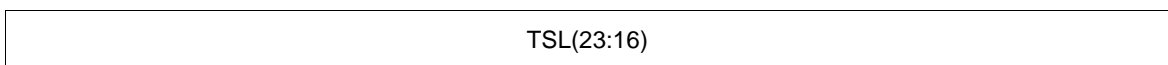
Access : read/write

Address : 08_H

Reset Value : 00FF_H

15

0



TSL Time slot 31..16 Select

Selected bits in bit field TSL and RPTSL0.TSL determine those time slots, which are used for PRBS monitoring. Time slots can be programmed arbitrarily. E.g. if RPTSL0.TSL(1) and RPTSL0.TSL(2) are set to '1', the PRBS is monitored continuously over both time slots combined.

IMR
T1/E1 Receive Interrupt Mask Register

Access : read/write
Address : 09_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|------|-----|----------|-----|------|------|------|-----|----|-----|------|-------|---|
| 15 | | | | | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | T400 | CRC | PDEN/AUX | FAS | MFAS | AISS | LOSS | RAS | ES | SEC | LLBS | PRBSS | |

For each framer interrupt vector an interrupt vector generation mask is provided. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see "Layer One Interrupts" on page 136.

The following definition applies:

- 1 The corresponding interrupt vector is suppressed by the device.
- 0 The corresponding interrupt vector is generated.

- T400 Mask '400 millisecond Timer'
- CRC Mask 'CRC Error'
- PDEN/AUX Mask 'Pulse Density / Auxiliary Pattern'
- FAS Mask 'Frame Alignment Status'
- MFAS Mask 'Multiframe Alignment Status'
- AIS Mask 'Alarm Indication Status'
- LOSS Mask 'Loss of Signal Status'
- RAS Mask 'Remote Alarm Status'
- ES Mask 'Errored Second'
- SEC Mask 'One Second Tick'
- LLBS Mask 'Line Loopback Status'
- PRBSS Mask 'PRBS Status'

RFMR1
T1/E1 Receive Mode Register 1

Access : read/write
Address : 0A_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------|------|-----|---|
| 15 | | | | | | | | | | | | | | 4 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | FRST(2:0) | EACM | ECM | |

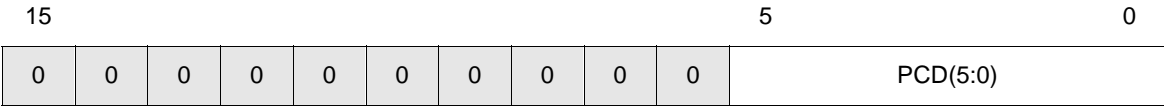
- FRST** Force Resynchronization Timer
This bit field defines the time after which the framer automatically starts resynchronization if Emulator Automatic Check Mode is enabled.
0..7 Automatic resynchronization after (FRST+1)*8 milliseconds.
- EACM** Enable Emulator Automatic Check Mode
This bit enables automatic resynchronization mode. After loss of frame the receive framer starts resynchronization after (FRST+1)*8ms when frame search is not started by system software. If EACM is disabled system software has to force resynchronization by setting bit RCMDR.FRS.
- ECM** Error Counter Mode
- 0 Unbuffered error counter mode. Counters are updated when respective error occurs. Counter registers are directly readable and cleared automatically at the end of a read cycle.
 - 1 Buffered error counter mode. Actual error counts are hidden from user and updated in background. The counter is copied to the bus register at one second intervals and reset automatically. This operation is synchronous with the periodic one second interrupt which alerts software to read the register.

PCD
T1/E1 Receive Pulse Count Detection Register

Access : read/write

Address : 0B_H

Reset Value : 0015_H



PCD Pulse Count Detection

A 'Loss of Signal' alarm will be detected, if the incoming data stream has zero octets for a programmable number T of consecutive octets. The number T is programmable via the PCD register and can be calculated as follows:

$$T = 8*(PCD+1), 1 ≤ PCD ≤ 63.$$

E.g. PCD = 21 sets loss of signal threshold to 176 (=(21+1)*8) zeros.

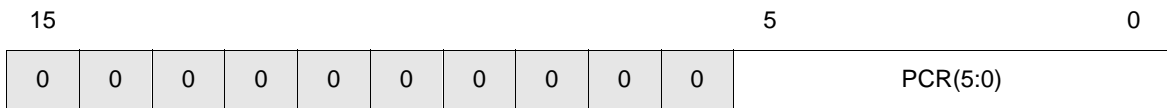
Note: For T1 mode time detection interval has cumulative uncertainty of 1 per 193 clocks.

PCR
T1/E1 Receive Pulse Count Recovery Register

Access : read/write

Address : 0C_H

Reset Value : 0015_H



PCR Pulse Count Recovery

'Loss of Signal' alarm will be cleared, when a programmable pulse density is detected in the received bit stream. A pulse is a logical '1' in the received bit stream. The number of pulses M which must occur in a certain time interval, which is programmable via register PCR, can be calculated as follows:

$M = PCR, 1 \leq PCR \leq 63.$

Additional 'Loss of Signal' recovery condition may be selected by using RFMR.LOSR.

FRS
T1/E1 Receive Status Register

Access : read/write
Address : 40_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | |
|----|-----|-----|-----|-----|-----|------|------|---|---|---|---|-------------|-------|-------|------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | | | | | 3 | 2 | 1 | 0 |
| 0 | NMF | LOS | AIS | LFA | RRA | LMFA | FSRF | 0 | 0 | 0 | 0 | PDEN AUX | LLBDD | LLBAD | PRBS |

Each bit in the framer receive status register declares a specific condition dependent on the selected modes. The following convention applies to the individual bits:

- 0 The named status is not or no longer existing.
- 1 The named status is currently effective.

The change of status bit (except FSRF) can also be used to generate a framer interrupt vector. See also register IMR which describes how to enable/disable interrupt vector generation and refer to the description of framer interrupt vector on page "Layer One Interrupts" on page 136.

NMF No Multiframe Found

E1: CRC-4 Interworking

This bit is set, if no multiframe is found after 400 milliseconds.

LOS Loss of Signal (Red Alarm)

This bit is set, when the 'Loss of Signal' condition has been detected.

T1

Detection

An alarm will be generated if the incoming data stream remain at logical zero for 168 cycles.

Recovery

The recovery procedure starts after detecting a logical 1. The LOS alarm is cleared if 21 one's are detected within 168 bits (12.5%).

E1

see T1 and "Error Performance Monitoring and Alarm Handling" on page 94.

Register Description

| | |
|------|--|
| AIS | <p>Alarm Indication Signal (AIS)</p> <p>This bit is set, when the alarm indication condition defined by bit RFMR.AIS3 has been detected. The flag stays active for at least one multiframe. It will be reset with the beginning of the next following multiframe, if no alarm condition is detected.</p> |
| LFA | <p>Loss of Frame Alignment</p> <p>T1</p> <p>This bit is set, when the 'Loss of Frame Alignment' condition defined by bits RFMR.SSP and RFMR.SSC has been detected. The flag is cleared, when synchronization has been regained.</p> <p>E1</p> <p>This bit is set, when the 'Loss of Frame Alignment' condition defined by bit RFMR.SSC has been detected. The flag is cleared, when synchronization has been regained.</p> |
| RRA | <p>Received Remote Alarm (Yellow Alarm)</p> <p>Condition for receive remote alarm is defined by bit FMR.RRAM. The flag is set after detecting remote alarm (yellow alarm).</p> |
| LMFA | <p>Loss of Multiframe Alignment</p> <p>T1: F12</p> <p>This bit is set, when the condition for 'Loss of Multiframe Alignment' defined by bit RFMR.SSC has been detected. The flag is cleared after multiframe synchronization has been regained.</p> <p>E1: CRC-4 Multiframe, CRC-4 Interworking</p> <p>This bit is set in CRC-4 multiframe or CRC-4 interworking mode, when double frame alignment is lost. This bit is reset, when the multiframe pattern is acquired or after 400 milliseconds in CRC-4 interworking mode, when NMF is asserted.</p> |
| FSRF | <p>Frame Search Restart Flag</p> <p>This bit toggles on each new pulse frame search started. This function can be used to recognize multiple candidates. If FSRF does not toggle, but LFA and LMFA remain active, the synchronizer has multiple candidates and cannot determine which one is correct.</p> <p><i>Note: This flag can not be used to generate an interrupt vector.</i></p> |

Register Description

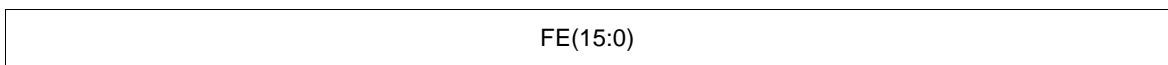
| | |
|----------|--|
| PDEN/AUX | <p>T1</p> <p>Pulse Density Code Violation Detected</p> <p>This bit is set, when the pulse density of the received data stream is below the requirement defined by ANSI T1.403.</p> <p>E1</p> <p>Auxiliary Pattern Detected</p> <p>This bit is set, when the pattern '...010101...' has been detected concurrent with loss of frame.</p> |
| LLBDD | <p>Line Loop-Back Deactuation Signal Detected</p> <p>This bit is set, when line loopback deactuate signal is detected and then received over a period of more than 33,16ms with a bit error rate less than 1/100. The bit remains set as long as the bit error rate does not exceed 1/100.</p> <p>If framing is aligned, the first bit position of any frame is not taken into account for the error rate calculation. If frame alignment state is not synchronized, all received data bits are searched for the LLBD pattern.</p> |
| LLBAD | <p>Line Loop-Back Actuation Signal Detected</p> <p>This bit is set to one in case the LLB actuate signal is detected and then received over a period of more than 33,16ms with a bit error rate less than 1/100. The bit remains set as long as the bit error rate does not exceed 1/100.</p> <p>If framing is aligned, the first bit position of any frame is not taken into account for the error rate calculation. If frame alignment state is not synchronized, all receive data bits are searched for the LLBA pattern.</p> |
| PRBS | <p>PRBS status</p> <p>This bit is set, when the PRBS receiver is in the synchronous state. It is set high if the synchronous state is reached even in the presence of a BER 1/10. A data stream containing all zeros with / without framing bits is also a valid pseudo-random bit sequence.</p> |

FEC
T1/E1 Receive Framing Error Counter

Access : read/write
Address : 41_H
Reset Value : 0000_H

15

0



FE Framing Error Counter

The counter will not be incremented during asynchronous state. Error counter mode (Clear on Read or Errored Second) depends on register RFMR1.ECM. In errored second mode the counter is 10 bit wide, otherwise 16 bit.

T1: F12

The counter will be incremented when incorrect FT and FS bits are received.

T1: ESF

The counter will be incremented when incorrect FAS bits are received.

E1

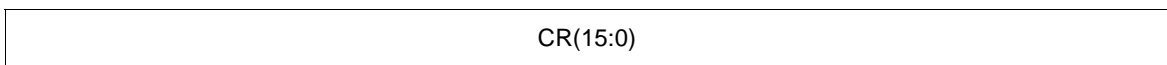
The counter will be incremented when incorrect FAS words are received.

CEC
T1/E1 Receive CRC Error Counter

Access : read/write
Address : 42_H
Reset Value : 0000_H

15

0



CR CRC Errors
The counter will not be incremented during asynchronous state. Error counter mode (Clear on Read or Errored Second) depends on register RFMR1.ECM. In errored second mode the counter is 10 bit wide, otherwise 16 bit.

T1: F12

No function.

T1: ESF

The counter will be incremented when a multiframe has been received with a CRC error.

E1: Doubleframe

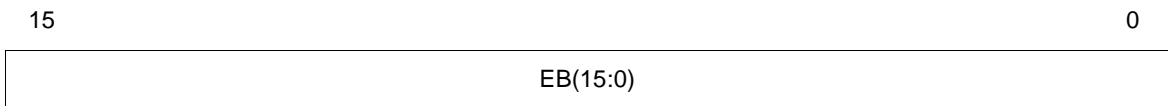
No function.

E1: CRC-4 Multiframe

In CRC-4 multiframe mode the counter will be incremented when a submultiframe has been received with a CRC error.

EBC
T1/E1 Receive Errored Block Counter

Access : read/write
Address : 43_H
Reset Value : 0000_H



EB E-Bit or Errored Block counter

The counter will not be incremented during asynchronous state. Error counter mode (Clear on Read or Errored Second) depends on register RFMR1.ECM. In errored second mode the counter is 10 bit wide, otherwise 16 bit.

T1

The counter will be incremented once per multiframe if a submultiframe has been received with a CRC error or an errored frame alignment has been detected.

E1: Doubleframe

No function.

E1: CRC-4 Multiframe

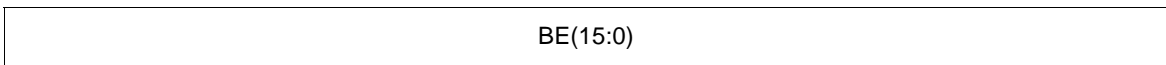
The counter will be incremented each time the framer receives a CRC-4 multiframe with S_i bit in frame 13 or frame 15 set to zero.

BEC
T1/E1 Receive Bit Error Counter

Access : read/write
Address : 44_H
Reset Value : 0000_H

15

0



BE Bit Error Counter
Error counter mode (Clear on Read or Errored Second) depends on register RFMR1.ECM. In errored second mode the counter is 10 bit wide, otherwise 16 bit.

T1

This bit counter will be incremented with every received PRBS bit error in the PRBS synchronous state.

8.2.7 Facility Data Link Registers

Facility data link registers control the signalling channels of T1, E1 as well as the signalling channels of the DS3 C-bit parity format (Path Maintenance Data Link and Far End Alarm and Control Channel).

RCR1 Receive Channel Configuration Register 1

Access : read/write

Address : 00_H

Reset Value : 0000_H

| | | | | | | | | | | | | | | | |
|----|------|------|----------|-----|-------|-----|-----|------|------|------|------|---------|-----|------|---|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | RAH2 | RAH1 | RTF(1:0) | INV | RIFTF | BFE | BRM | BRAC | RAL2 | RAL1 | XCRC | CRC DIS | RON | HDLC | |

RAH2 Receive Address High Byte 2 Valid

This bit enables byte RAH.RAH2 for address comparison.

0 Disable

1 Enable

RAH1 Receive Address High Byte 1 Valid

This bit enables byte RAH.RAH1 for address comparison.

0 Disable

1 Enable

RTF RFIFO Threshold Level

This bit field sets the threshold of the receive FIFO and is applied to both pages of the receive FIFO. A 'Receive Pool Full' interrupt vector will be generated, when the programmed threshold is reached. The threshold value is given as follows:

00_B 32 byte threshold

01_B 16 byte threshold

10_B 4 byte threshold

11_B 2 byte threshold

Register Description

| | |
|-------|---|
| INV | <p>Invert data input from Receive Framer</p> <p>This bit enables data inversion between receive framer and receive signalling controller.</p> <p>0 Disable data Inversion.</p> <p>1 Enable data inversion.</p> |
| RIFTF | <p>Report Interframe Time-fill Change</p> <p>This bit selects, that interframe time-fill changes should be reported.</p> <p>0 Disable IFF status messages.</p> <p>1 Enable IFF status messages.</p> |
| BFE | <p>Enable BOM Filter Mode</p> <p>This bit selects, that byte oriented messages have to be filtered. The BOM is reported only if 7 out 10 data is received. This bit is valid in BOM mode only.</p> <p>0 Disable BOM filter mode.</p> <p>1 Enable BOM filter mode.</p> |
| BRM | <p>BOM Receive Mode</p> <p>This bit switches continuous and 10 byte packet reception of the receive signalling controller. This bit is valid in BOM mode only.</p> <p>0 Enable continuous reception.</p> <p>1 Enable 10 bytes packets.</p> |
| BRAC | <p>BOM Receiver Active</p> <p>T1: ESF</p> <p>This bit switches the BOM receiver to operational state (on) or inoperational state (off). When BOM Receiver is switched on, an automatic switching between HDLC mode and BOM mode is enabled. If eight or more consecutive '1's are detected, the BOM mode is entered. Upon detection of a flag in the data stream, the signalling controller switches back to HDLC mode.</p> <p>0 Switch BOM receiver off.</p> <p>1 Switch BOM receiver on.</p> |
| RAL2 | <p>Receive Address Low Byte 2 Valid</p> <p>This bit enables byte RAL.RAL2 for address comparison.</p> <p>0 Disable</p> <p>1 Enable</p> |

Register Description

| | |
|--------|--|
| RAL1 | Receive Address Low Byte 1 Valid This bit enables byte RAL.RAL1 for address comparison. 0 Disable 1 Enable |
| XCRC | Transfer CRC to RFIFO This bit defines, that CRC of incoming data packets shall be transferred to the receive FIFO or not. 0 No transfer of CRC to RFIFO. 1 Transfer of CRC to RFIFO. |
| CRCDIS | CRC Check Disable This bit enables or disables the CRC check of incoming data packets. 0 Enable CRC check. 1 Disable CRC check. |
| RON | Receiver On/Off This bit switches the receiver of the facility data link channel to operational (on) or inoperational state (off). 0 Switch receiver off. 1 Switch receiver on. |
| HLDC | HDLC Mode This bit identifies the protocol mode of the facility data link receiver. 0 Set protocol mode to transparent. 1 Set protocol mode to HDLC. |

RCR2 Receive Channel Configuration Register 2

Access : read/write
Address : 01_H
Reset Value : 0000_H

| | | | | | | | | | | | | | |
|-----|------|------|------------|------------|------|------|------|------|------|-----|------|---|---|
| 15 | 14 | 13 | 12 | 10 | 9 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PAS | SAUM | SAUP | SACRC(2:0) | SASSM(2:0) | SA8E | SA7E | SA6E | SA5E | SA4E | SMF | T1E1 | | |

- PAS** Pattern Select for SSM and CRC Count Function
This bit selects the default pattern for synchronization status messages and bit error indication.
0 Use pattern defined in ETS 300233.
1 Use patterns specified in registers VSSM and VCRC.
- SAUM** S_a-bit Update Mode
This bit selects the update mode for the S_a-bits located in register RSAW1..RSAW3.
E1: Doubleframe
0 S_a-bits are updated after eight frames.
1 S_a-bits are updated only, if S_a data changes. Update is done after eight frames.
E1: CRC-4 Multiframe
0 S_a-bits are updated after every multiframe.
1 S_a-bits are updated only, if S_a data changes. Update is done on a multiframe start.
- SAUP** S_a-Bit Update
This bit enables the S_a-bit update function.
0 Disable update of S_a-bits.
1 Enable update of S_a-bits using RSAW1..RSAW3 registers.

Register Description

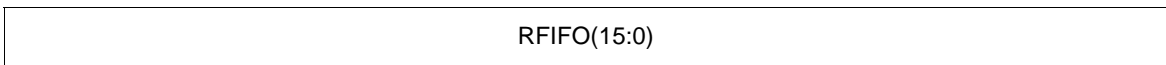
| | |
|------------|---|
| SACRC | <p>S_a-bit Select for CRC Function</p> <p>This bit field enables the CRC count function of the selected S_a-bit.</p> <p>0 Disable CRC count function.</p> <p>1..5 Enable CRC count function for bit $S_{a4}..S_{a8}$, e.g. SACRC = 2 selects bit S_{a8} for CRC count function.</p> <p>Other Reserved</p> |
| SASSM | <p>S_a-bit Select for SSM Function</p> <p>This bit field enables the synchronization status message function of the selected S_a-bit. The SSM function checks incoming messages and reports any change if a synchronization status message has been received three times in a row.</p> <p>0 Disable SSM function.</p> <p>1..5 Enable SSM function for bit $S_{a4}..S_{a8}$, e.g. SASSM = 2 selects bit S_{a8} for SSM function.</p> <p>Other Reserved</p> |
| SA8E..SA4E | <p>S_a-bit Signalling Enable</p> <p>Setting one of the bits switches between S_a-bit access or protocol access of the selected bits.</p> <p>0 Enable S_a-bit access via register RSAW1-3.</p> <p>1 Enable protocol access (HDLC, transparent). Selected bits will be combined to receive protocol data.</p> |
| SMF | <p>Select Multiframe Format</p> <p>This bit switches between doubleframe and CRC-4 multiframe format.</p> <p>0 Select doubleframe format.</p> <p>1 Select CRC-4 multiframe format.</p> |
| T1E1 | <p>T1/E1 Mode Selection</p> <p>This bit switches the receive signalling controller into T1 or E1 mode.</p> <p>0 Select T1 mode.</p> <p>1 Select E1 mode.</p> |

RFF
Receive FIFO Register

Access : read
Address : 02_H
Reset Value : 0000_H

15

0



RFIFO **Receive FIFO Data**

This bit field contains the first 16 bit word of the receive FIFO of the signalling controller. The receive FIFO itself consists of two pages with 32 bytes, thus 16 words can be stored inside the receive FIFO at a time. Port status and FIFO operations can be accessed via register PSR and register HND.

The first bit received is stored in bit 0.

XCR1
Transmit Channel Configuration Register 1

Access : read/write
Address : 03_H
Reset Value : 0000_H



- PBYTE** Preamble Byte
This bit field selects the preamble byte to be sent after interframe time-fill transmission is stopped.
- PCNT** Preamble Count
This bit field selects the amount of preamble repetitions.
- INV** Invert Data
This bit enables data inversion between transmit signalling controller and transmit framer.
0 Disable data Inversion.
1 Enable data inversion.
- XON** Transmitter On/Off
This bit switches the transmitter of the facility data link to operational (on) or inoperational state (off).
0 Switch transmitter off.
1 Switch transmitter on.
- DISCRC** Disable CRC
This bit enables CRC generation and transmission on transmission of HDLC packets.
0 Enable CRC generation.
1 Disable CRC generation.

Register Description

| | |
|----|---|
| SF | Shared Flags |
| | This bit enables transmission of protocol data with shared flags. |
| 0 | Disable shared flags. |
| 1 | Enable shared flags. |

XCR2 Transmit Channel Configuration Register 2

Access : read/write

Address : 04_H

Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|------|------|------|------|------|------|-----|------|
| 15 | | | | | | | | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | IFTF | SA8E | SA7E | SA6E | SA5E | SA4E | SMF | T1E1 |

IFTF Interframe Time Fill

This bit determines the interframe time of the transmit signalling controller.

0 Interframe time fill is 7E_H.

1 Interframe time fill is FF_H.

SA8E..SA4E S_a-bit Signalling Enable

Setting one of the bits switches between normal S_a-bit access or protocol access of the selected bits.

0 Enable S_a-bit access via register XSAW1-3.

1 Enable protocol access (HDLC, transparent). Selected bits will be combined for protocol data transmission.

SMF Select CRC-4 Multiframe Format

This bit switches between doubleframe and multiframe format.

E1

0 Select doubleframe format.

1 Select CRC-4 multiframe format.

T1E1 T1/E1 Mode Selection

This bit switches the receive signalling controller into T1 or E1 mode.

0 Select T1 mode.

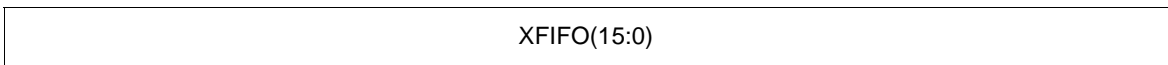
1 Select E1 mode.

XFF
Transmit FIFO Register

Access : write
Address : 05_H
Reset Value : 0000_H

15

0



XFIFO **Transmit FIFO Data**

This bit field writes a 16 bit word to the transmit FIFO of the signalling controller. The transmit FIFO itself consists of two pages with 32 bytes, thus 16 words can be written to the transmit FIFO at a time. Port status and FIFO operations can be accessed via register PSR and register HND.

Data written to the transmit FIFO is sent starting with bit 0 up to bit 15.

PSR
Port Status register

Access : read
Address : 06_H
Reset Value : 0000_H

| | | | | | | | | | |
|----|-----|-----|----------|------------|------|-----------|---|---|---|
| 15 | 14 | 13 | 12 | 8 | 7 | 6 | 5 | 4 | 0 |
| | XRA | XFW | RBC(4:0) | SMODE(1:0) | BRFO | STAT(4:0) | | | |

- XRA** Transmit Repeat Active
This bit indicates that the transmit signalling controller is operating in repeat mode.
0 Normal operation
1 Repeat operation
- XFW** Transmit FIFO Write Enable
This bit indicates that data can be written to XFF.XFIFO. This bit is for polling use with the same meaning as the 'Transmit Pool Ready' interrupt vector.
- RBC** Receive Byte Count
This bit field indicates the amount of data stored in the receive FIFO. Valid after a 'Receive Message End' interrupt vector is generated. Receive byte count will be cleared, when a 'Receive Message Clear' command is executed via register HND. A zero byte count in combination with a 'Receive Pool Full' or 'Receive Message End' interrupt vector means that 32 bytes are available in the receive FIFO.
- SMODE** Receiver Status Mode
This bit indicates the status of the receive signalling controller. If BOM mode is selected via bit RCR1.BRM the receiver switches automatically between HDLC mode and BOM mode.
10_B HDLC mode
01_B BOM mode
Other Reserved

Register Description

| | |
|------|--|
| BRFO | <p>BOM Receive FIFO Overflow</p> <p>0 No overflow</p> <p>1 RFF overflow</p> <p>The status word will be cleared after a 'Receive Message Clear' command is issued.</p> |
| STAT | <p>Receive FIFO Status</p> <p>This bit field reports the status of the data stored in the receive FIFO.</p> <p>HDLC mode</p> <p>00000_B Valid HDLC Frame</p> <p>00001_B Receive Data Overflow</p> <p>00010_B Receive Abort</p> <p>00011_B Not Octet</p> <p>00100_B CRC Error</p> <p>00101_B Channel Off</p> <p>BOM MODE</p> <p>00000_B BOM Filtered data declared</p> <p>00001_B BOM data available</p> <p>00010_B BOM End</p> <p>00011_B BOM filtered data undeclared</p> <p>00100_B BOM header error (ISF, incorrect synchronization format)</p> |

HND Handshake Register

Access : write
Address : 07_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|-----|---|-------|------|------|-----|-----|-----|-----|
| 15 | | | | | | | | 8 | | | | | | | | |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | RMC | 0 | ABORT | XRES | XREP | OBI | XHF | XTF | XME |

Note: Receive command (bit 8) and transmit commands (bit 5 down to bit 0) can not be issued at the same time. Doing so will cause the facility data link to omit the transmit commands.

- RMC** Receive Message Complete
This bit is a confirmation from CPU that a data block has been read from RFIFO following a 'Receive Pool Full' or 'Receive Message End' interrupt vector and that the occupied page can now be released.
0 No function
1 Release page of receive FIFO.
Note: If this bit is set, the low byte (transmit commands) of the register HND is ignored.
- ABORT** Abort Frame
Setting this bit aborts HDLC frames which are transmitted.
0 Normal operation
1 Abort HDLC frame.
- XRES** Transmitter Reset
This bit resets the signalling controller transmit. However, the contents of the control register will not be reset.
0 Normal operation
1 Transmitter reset
- XREP** Transmission Repeat
Setting this bit together with bit XTF indicates that the contents stored in XFF.XFIFO shall be repeatedly transmitted by the MUNICH256FM.
0 No cyclic transmission.

Register Description

| | | |
|-----|---|--|
| | 1 | Enable cyclic transmission. |
| OBI | | Odd Byte Count Indicator |
| | | Setting this bit together with bit XME indicates the number of bytes written to XFF.XFIFO is odd. This means the <u>lower</u> byte of the last write transfer to the transmit FIFO is valid only. In HDLC mode the status byte written to transmit FIFO must be included in calculation. |
| | 0 | Even number of bytes stored in XFF.XFIFO. |
| | 1 | Odd number of bytes stored in XFF.XFIFO. |
| XHF | | Transmit HDLC frame |
| | | Setting this bit indicates that the contents written to XFF.XFIFO shall be transmitted as HDLC frame. If data written to XFF.XFIFO completes a HDLC frame, bit XME must be set together with XHF in order to generate CRC and flag. |
| | 0 | No function |
| | 1 | Transmit data stored in XFF.XFIFO in HDLC format. |
| XTF | | Transmit transparent frame |
| | | Setting this bit indicates that the contents written to XFF.XFIFO shall be transmitted in transparent mode. |
| | 0 | No function |
| | 1 | Transmit data stored in XFF.XFIFO fully transparent, i.e. without bit stuffing and CRC. |
| XME | | Transmit Message End |
| | | Setting this bit indicates that the last data block written to XFF.XFIFO completes the current frame. The last byte of the data block written to the transmit FIFO is a status word indicating the message status. The signalling controller terminates the transmission properly by appending CRC and the closing flag to the data sequence if the status word written as the last entry to the transmit FIFO does not contain an abort indication. |

Register Description

Table 8-8 Signalling Controller Transmit Commands

| XRES | XREP | OBI | XHF | XTF | XME | Function |
|------|------|-----|-----|-----|-----|---|
| 1 | - | - | - | - | - | Reset Port |
| 0 | 0 | 0 | 1 | 0 | 0 | Transmit HDLC Frames Send FIFO content as HDLC frame. |
| 0 | 0 | 0/1 | 1 | 0 | 1 | End Transmit HDLC Send FIFO content as HDLC frame. Add CRC (if enabled) and flag after last byte stored in FIFO. |
| 0 | 1 | 0/1 | 1 | 0 | 0 | Repeat HDLC Frame Send FIFO content as HDLC frame. Add CRC (if enabled) and flag after last byte stored in FIFO. Then repeat transmission of FIFO content. |
| 0 | 1 | 0/1 | 1 | 0 | 1 | Stop Repeat HDLC Frame Stop transmission after last byte stored in FIFO. This command is issued when repetitive transmission started by command 'Repeat HDLC Frame' shall be stopped. |
| 0 | 0 | 0 | 0 | 1 | 0 | Transmit Transparent Send FIFO content in transparent mode. |
| 0 | 0 | 0/1 | 0 | 1 | 1 | End Transmit Transparent Send FIFO content in transparent mode. End transmission after last byte stored in FIFO. |
| 0 | 1 | 0/1 | 0 | 1 | 0 | Repeat Transmit Transparent Send FIFO content in transparent mode. Repeat transmission of FIFO content after last byte was sent. |
| 0 | 1 | 0/1 | 0 | 1 | 1 | Stop Repeat Transmit Transparent Stop transparent transmission after last byte stored in FIFO. This command is issued when repetitive transmission started by command 'Repeat transmit transparent' shall be stopped. |

MSK
Interrupt Mask Register

Access : read/write
Address : 08_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|------|------|-----|-----|---|---|---|---|-----|-----|-----|-----|-----|
| 15 | | | | | 11 | 10 | 9 | 8 | | | | | 4 | 3 | 2 | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | TXSA | ALLS | XDU | XPR | 0 | 0 | 0 | 0 | RSA | SSM | RPF | RME | ISF |

For each facility data link interrupt vector an interrupt vector generation mask is provided. Generation of an interrupt vector itself does not necessarily result in assertion of the interrupt pin. For description of interrupt concept and interrupt vectors see "Layer One Interrupts" on page 136.

The following definition applies:

- 1 The corresponding interrupt vector will not be generated by the device.
- 0 The corresponding interrupt vector will be generated.

Facility Data Link Interrupt Vector Transmit

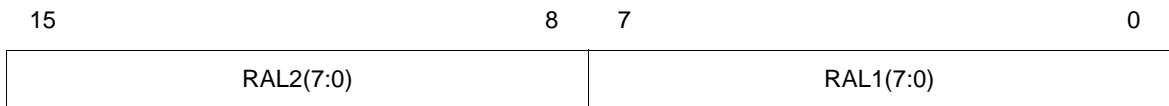
- TXSA Mask 'Transmit S_a Data'
- ALLS Mask 'All Sent'
- XDU Mask 'Transmit Data Underrun'
- XPR Mask 'Transmit Pool Ready'

Facility Data Link Interrupt Vector Receive

- RSA Mask 'Receive S_a Data Valid'
- SSM Mask 'Synchronization Status Message Received'
- RPF Mask 'Receive Pool Full'
- RME Mask 'Receive Message End'
- ISF Mask 'Incorrect Synchronization Format'

RAL
Receive Address Low

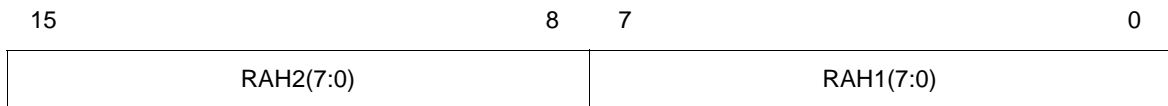
Access : read/write
Address : 09_H
Reset Value : 0000_H



- RAL2** Receive Address Low Byte
This bit field defines the low byte of the second receive address.
- RAL1** Receive Address Low Byte
This bit field defines the low byte of the first receive address.

RAH
Receive Address High

Access : read/write
Address : 0A_H
Reset Value : 0000_H

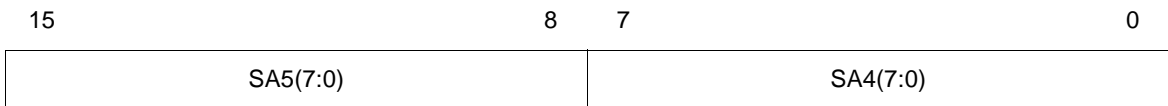


RAH2 Receive Address High Byte
This bit field defines the high byte of the second receive address.

RAH1 Receive Address High Byte
This bit field defines the high byte of the first receive address.

RS AW1
Receive S_a Word 1

Access : read
Address : 0B_H
Reset Value : 0000_H



SA5 Received S_{a5} Data Byte
This bit field contains data received in S_{a5} of an E1 doubleframe or an E1 CRC-4 multiframe.

E1: CRC-4 Multiframe

Received data byte is aligned to a multiframe boundary. SA5(0) is the data bit receive in frame one, while SA5(7) is the data byte received in frame 15 of a multiframe.

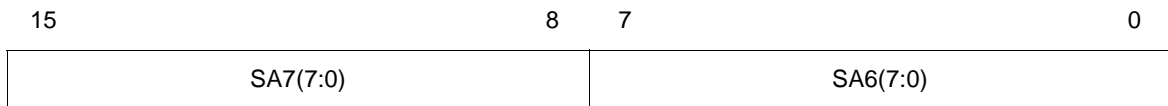
SA4 Received S_{a4} Data Byte
This bit field contains data received in S_{a4} of an E1 doubleframe or an E1 multiframe.

E1: CRC-4 Multiframe

Received data byte is aligned to a multiframe boundary. SA4(0) is the data bit receive in frame one, while SA4(7) is the data byte received in frame 15 of a multiframe.

RS AW2
Receive S_a Word 2

Access : read
Address : 0C_H
Reset Value : 0000_H



SA7 Received S_{a7} Data Byte
This bit field contains data received in S_{a7} of an E1 doubleframe or an E1 CRC-4 multiframe.

E1: CRC-4 Multiframe

Received data byte is aligned to a multiframe boundary. SA7(0) is the data bit receive in frame one, while SA7(7) is the data byte received in frame 15 of a multiframe.

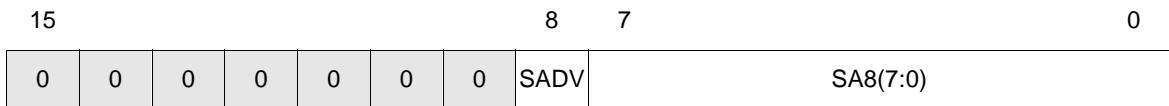
SA6 Received S_{a6} Data Byte
This bit field contains data received in S_{a6} of an E1 doubleframe or an E1 multiframe.

E1: CRC-4 Multiframe

Received data byte is aligned to a multiframe boundary. SA6(0) is the data bit receive in frame one, while SA6(7) is the data byte received in frame 15 of a multiframe.

RSAW3
Receive S_a Word 3

Access : read
Address : 0D_H
Reset Value : 0000_H



SADV Received S_{a4}..S_{a8} Data Valid
This bit indicates that new S_a data in register RSAW1..RSAW3 is available. The signalling controller will not update S_a data while this bit is set. SADV will be cleared on reads to this register.

- 0 No S_a data available.
- 1 S_a data available in register RSAW1..RSAW3.

SA8 Received S_{a8} Data Byte
This bit field contains data received in S_{a8} of an E1 doubleframe or an E1 multiframe.

E1: CRC-4 Multiframe

Received data byte is aligned to a multiframe boundary. SA8(0) is the data bit receive in frame one, while SA8(7) is the data byte received in frame 15 of a multiframe.

RSAW4
Receive S_a Word 4

Access : read
Address : 0E_H
Reset Value : 0000_H

| | | | | | | | | | | | | | | | | | | |
|----|---|---|---|---|---|---|---|---|-----------|---|---|---|---|---|---|---|---|------|
| 15 | | | | | | | | | 7 | | | | | 4 | 3 | | 1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSMD(3:0) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | SSMV |

SSMD SSM Data Pattern

This bit field contains the received synchronization status message. The synchronization status message reported depends on bit RCR2.PAS and, if selected, on pattern enabled in register VSSM. Only valid if SSMV is set.

SSMV Synchronization Status Message Valid

This bit indicates that a new synchronization status message has been received. A new SSM is reported every time a message has been received three time in a row on the S_a-bit selected via register RCR2.SASSM. This bit is reset after the user performs a read on this register.

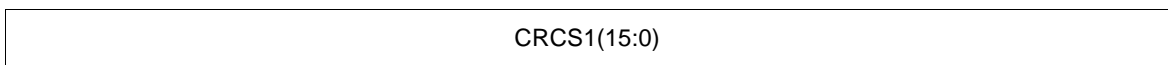
- 0 No new SSM data available.
- 1 New SSM data available.

CRC1
CRC Status Counter 1

Access : read
Address : 0F_H
Reset Value : 0000_H

15

0



CRC1 CRC1 counter

The S_a-bit error indication counter CRC1 (16 bits) counts either the received bit sequences 0001_B and 0011_B or user programmable values specified in register VCRC in every submultiframe on a selectable S_a-bit. In the primary rate access digital section CRC errors are reported from the TE via S_{a6}. Incrementing is only possible in the multiframe synchronous state.

The counter is increased with every received bit error indication if enabled in register RCR2. The counter will not be incremented once it reaches FFFF_H. A read will clear this counter.

CRC2
CRC Status Counter 2

Access : read
Address : 10_H
Reset Value : 0000_H

15

0

CRC2(15:0)₂

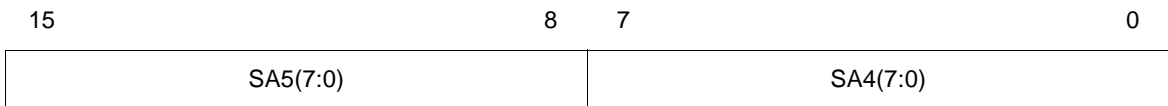
CRC2 CRC2 counter

The S_a-bit error indication counter CRC2 (16 bits) counts either the received bit sequences 0010_B and 0011_B or user programmable values specified in register VCRC in every submultiframe on a selectable S_a-bit. In the primary rate access digital section CRC errors detected at T-reference points are reported via S_{a6}. Incrementing is only possible in the multiframe synchronous state.

The counter is increased with every received bit error indication if enabled in register RCR2. The counter will not be incremented once it reaches FFFF_H. A read will clear this counter.

XSAW1
Transmit S_a Word 1

Access : read/write
Address : 11_H
Reset Value : 0000_H



SA5 Transmit S_{a5} Data Byte
This bit field contains data to be transmitted in S_{a5} of an E1 doubleframe or an E1 CRC-4 multiframe. SA5 will be inserted into the data stream, if selected via bit XCR2.SA5E.

E1: CRC-4 Multiframe

Transmit data will be aligned to a multiframe boundary. SA5(0) is the data bit transmitted in frame one while SA5(7) is the data bit transmitted in frame 15 of a multiframe.

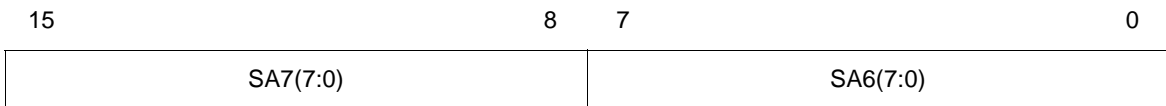
SA4 Transmit S_{a4} Data Byte
This bit field contains data to be transmitted in S_{a4} of an E1 doubleframe or an E1 CRC-4 multiframe. SA4 will be inserted into the data stream, if selected via bit XCR2.SA4E.

E1: CRC-4 Multiframe

Transmit data will be aligned to a multiframe boundary. SA4(0) is the data bit transmitted in frame one while SA4(7) is the data bit transmitted in frame 15 of a multiframe.

XSAW2
Transmit S_a Word 2

Access : read/write
Address : 12_H
Reset Value : 0000_H



SA7 Transmit S_{a7} Data Byte

This bit field contains data to be transmitted in S_{a7} of an E1 doubleframe or an E1 multiframe. SA7 will be inserted into the data stream, if selected via bit XCR2.SA7E.

E1: CRC-4 Multiframe

Transmit data will be aligned to a multiframe boundary. SA7(0) is the data bit transmitted in frame one while SA7(7) is the data bit transmitted in frame 15 of a multiframe.

SA6 Transmit S_{a6} Data Byte

This bit field contains data to be transmitted in S_{a6} of an E1 doubleframe or an E1 CRC-4 multiframe. SA6 will be inserted into the data stream, if selected via bit XCR2.SA6E.

E1: CRC-4 Multiframe

Transmit data will be aligned to a multiframe boundary. SA6(0) is the data bit transmitted in frame one while SA6(7) is the data bit transmitted in frame 15 of a multiframe.

XSAW3
Transmit S_a Word 3

Access : read/write
Address : 13_H
Reset Value : 0000_H

| | | | | | |
|----|------|-----------|----------|---|---|
| 15 | 14 | 13 | 8 | 7 | 0 |
| 0 | XSAV | XSAR(5:0) | SA8(7:0) | | |

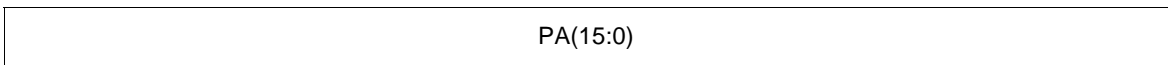
- XSAV** S_a Data Valid
This bit indicates that new S_a data has been written to register XSAW1..XSAW3 from system processor.
0 No new S_a data available.
1 New S_a data available.
- XSAR** S_a Data Repetitions
This bit field defines the number of repetitions of the S_a data bytes. A 'Transmit S_a Data' interrupt vector will be generated after programmed number of repetitions.
- SA8** Transmit S_{a8} Data Byte
This bit field contains data to be transmitted in S_{a8} of an E1 doubleframe or an E1 CRC-4 multiframe. SA8 will be inserted into the data stream, if selected via bit XCR2.SA8E.
E1: CRC-4 Multiframe
Transmit data will be aligned to a multiframe boundary. SA8(0) is the data bit transmitted in frame one while SA8(7) is the data bit transmitted in frame 15 of a multiframe.

VSSM
Valid SSM Pattern

Access : read/write
Address : 14_H
Reset Value : 0000_H

15

0



PA Pattern 15..0

Setting one or more of the bits enables the selected pattern for SSM comparison. E.g. setting PA(3) and PA(1) enables pattern 0010_B and 0001_B for SSM comparison. Identified SSM pattern are reported via register RSAW4.

Only valid if RCR2.PAS is set to '1'.

VCRC
Valid CRC Count Pattern

Access : read/write
Address : 15_H
Reset Value : 0000_H

| | | | | | | | |
|------------|----|------------|---|------------|---|------------|---|
| 15 | 12 | 11 | 8 | 7 | 4 | 3 | 0 |
| CRC22(3:0) | | CRC21(3:0) | | CRC12(3:0) | | CRC11(3:0) | |

CRC22

CRC21 CRC2 Pattern Definition

The bit fields CRC21 and CRC22 determine the S_a-bit error indication pattern to be reported in register CRC2.
Only valid if RCR2.PAS is set to '1'.

CRC12

CRC11 CRC1 Pattern Definition

The bit fields CRC11 and CRC12 determine the S_a-bit error indication pattern to be reported in register CRC1.
Only valid if RCR2.PAS is set to '1'.

9 Electrical Characteristics

9.1 Important Electrical Requirements

Both V_{DD3} and V_{DD25} can take on any power-on sequence. Within 50 milliseconds of power-up the voltages must be within their respective absolute voltage limits. At power-down, within 50 milliseconds of either voltage going outside its operational range, both voltages must be returned below 0.1V.

9.2 Absolute Maximum Ratings

Table 9-1 Absolute Maximum Ratings

| Parameter | Symbol | Limit Values | | Unit |
|--|-----------|--------------|---------------|------|
| | | min | max | |
| Ambient temperature under bias PEB 20256M E PEF 20256M E | T_A | 0 -40 | 70 85 | °C |
| Junction temperature under bias | T_J | | 125 | °C |
| Storage temperature | T_{stg} | -65 | 125 | °C |
| Voltage on any pin with respect to ground | V_S | -0.4 | $V_{DD3}+0.4$ | V |

Note: Stresses above those listed here may cause permanent damage to the device. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

9.3 DC Characteristics

a) Power Supply Pins

Table 9-2 DC Characteristics

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|---------------------|------------|--------------|------|------|----------------|
| | | min. | max. | | |
| Core Supply Voltage | V_{DD25} | 2.25 | 2.75 | V | |
| I/O Supply Voltage | V_{DD3} | 3.0 | 3.6 | V | |

Electrical Characteristics

| Parameter | | Symbol | Limit Values | | Unit | Test Condition |
|--|------------------------|----------------------|--------------|-------|---------|--|
| | | | min. | max. | | |
| Core supply current V_{DD25} | operational | I_{CC25} | | < tbd | mA | |
| | power down (no clocks) | I_{CCPD25} | | < 2 | mA | |
| I/O supply current V_{DD3} | operational | I_{CC3} | | < tbd | mA | Inputs at V_{SS}/V_{DD3} No output loads. |
| | power down (no clocks) | I_{CCPD3} | | < 2 | mA | |
| Sum of Input leakage current and Output leakage current (Outputs Hi-z) | | I_{LI} I_{LO} | | < 10 | μ A | |
| Power Dissipation | | P | | <3 | W | |

b) Non-PCI Interface Pins

Table 9-3 DC Characteristics (Non-PCI Interface Pins)

$T_A = -40$ to 85°C , $V_{DD3} = 3.3\text{ V} \pm 0.3\text{ V}$, $V_{DD25} = 2.5\text{ V} \pm 0.25\text{ V}$, $V_{SS} = 0\text{ V}$

| Parameter | | Symbol | Limit Values | | Unit | Test Condition |
|------------------|--|----------|--------------|---------------|------|------------------------------|
| | | | min. | max. | | |
| L-input voltage | | V_{IL} | -0.4 | 0.8 | V | |
| H-input voltage | | V_{IH} | 2.0 | $V_{DD3}+0.4$ | V | |
| L-output voltage | | V_{OL} | | 0.45 | V | $I_{QL} = 2\text{ mA}$ |
| H-output voltage | | V_{OH} | 2.4 | | V | $I_{QH} = -400\ \mu\text{A}$ |

c) PCI Interface Pins

Table 9-4 DC Characteristics (PCI Interface Pins)

$T_A = -40$ to 85°C , $V_{DD3} = 3.3\text{ V} \pm 0.3\text{ V}$, $V_{DD25} = 2.5\text{ V} \pm 0.25\text{ V}$, $V_{SS} = 0\text{ V}$

| Parameter | | Symbol | Limit Values | | Unit | Test Condition |
|-----------------|--|----------|--------------|---------------|------|----------------|
| | | | min. | max. | | |
| L-input voltage | | V_{IL} | -0.5 | $0.3V_{DD3}$ | V | |
| H-input voltage | | V_{IH} | $0.5V_{DD3}$ | $V_{DD3}+0.5$ | V | |

| Parameter | Symbol | Limit Values | | Unit | Test Condition |
|------------------|----------|--------------|--------------|------|-----------------------|
| | | min. | max. | | |
| L-output voltage | V_{OL} | | $0.1V_{DD3}$ | V | $I_{QL} = 1500 \mu A$ |
| H-output voltage | V_{OH} | $0.9V_{DD3}$ | | V | $I_{QH} = -500 \mu A$ |

9.4 AC Characteristics

a) Non-PCI interface pins

$T_A = -40$ to $85^\circ C$, $V_{DD3} = 3.3 V \pm 0.3 V$, $V_{DD25} = 2.5 V \pm 0.25 V$, $V_{SS} = 0 V$

Inputs are driven to 2.4 V for a logical '1' and to 0.4 V for a logical '0'. Timing measurements are made at 2.0 V for a logical '1' and at 0.8 V for a logical '0'.

The AC testing input/output waveforms are shown below.

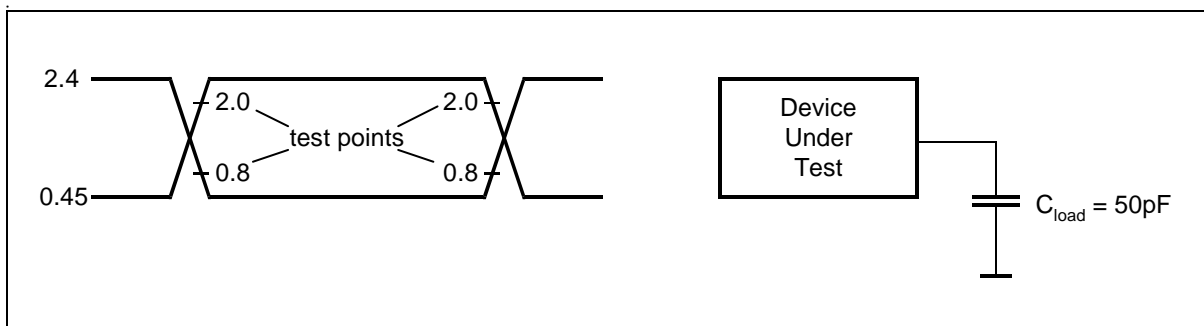


Figure 9-1 Input/Output Waveform for AC Tests

b) PCI interface pins

PCI interface pins are measured as pins compliant to the 3.3V signalling environment according to the PCI Specification Rev. 2.1.

9.4.1 PCI Bus Interface Timing

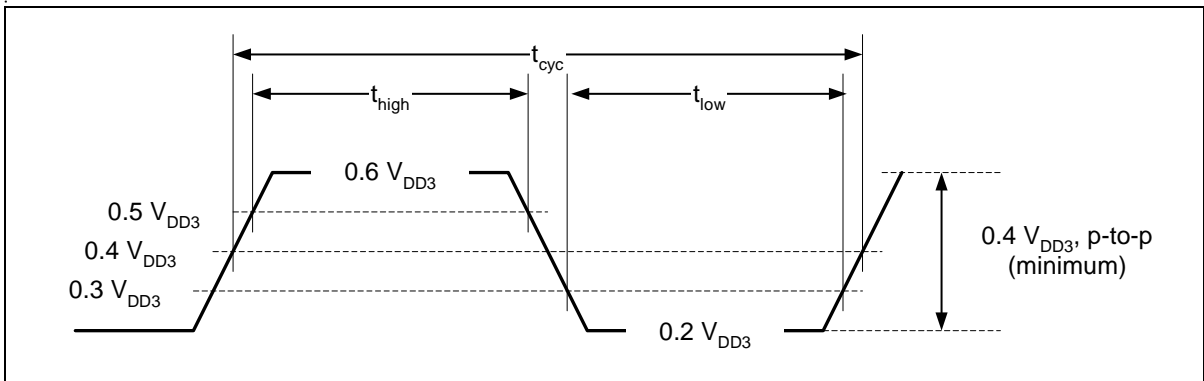


Figure 9-2 PCI Clock Cycle Timing

Table 9-5 PCI Clock Characteristics

| Parameter | Symbol | Limit Values | | Unit |
|--------------------------|------------|--------------|------|------|
| | | min. | max. | |
| CLK cycle time | t_{cyc} | 15 | | ns |
| CLK high time | t_{high} | 6 | | ns |
| CLK low time | t_{low} | 6 | | ns |
| CLK slew rate (see note) | | 1.5 | 4 | V/ns |

Note: Rise and fall times are specified in terms of the edge rate measured in V/ns. This slew rate must be met across the minimum peak-to-peak portion of the clock waveform shown in **Figure 9-3**.

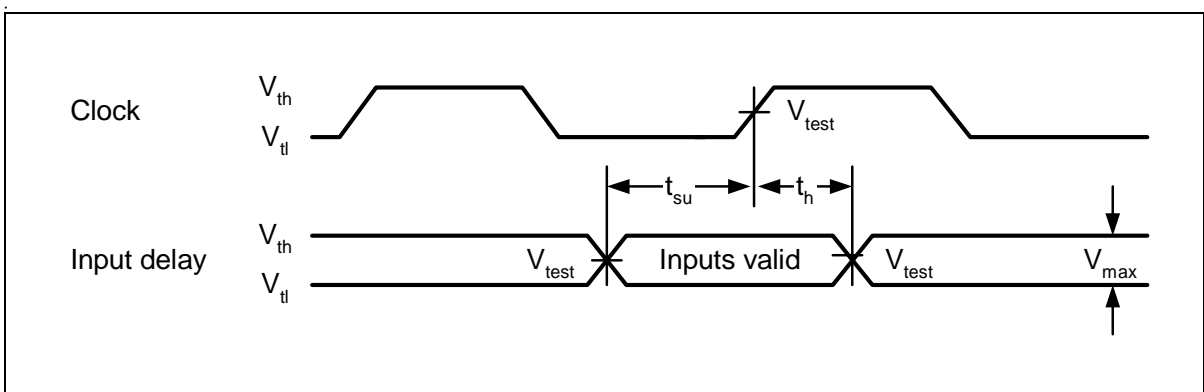


Figure 9-3 PCI Input Timing Measurement Conditions

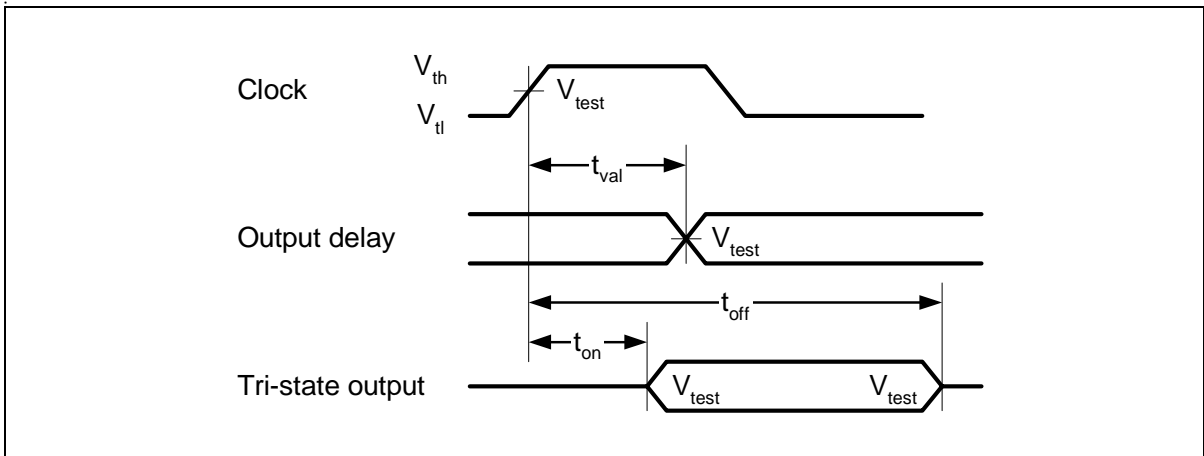


Figure 9-4 PCI Output Timing Measurement Conditions

Table 9-6 PCI Interface Signal Characteristics

| Parameter | Symbol | Limit Values | | Unit | Notes |
|--|-----------|--------------|------|------|-------|
| | | min. | max. | | |
| CLK to signal valid - bussed signals | t_{val} | 2 | 6 | ns | 1, 2 |
| CLK to \overline{REQ} valid | t_{val} | 2 | 6 | ns | 1, 2 |
| Float to active delay | t_{on} | 2 | | ns | |
| Active to float delay | t_{off} | | 14 | | |
| Input setup time to CLK - bussed signals | t_{su} | 3 | | | 2 |
| Input setup time to CLK - \overline{GNT} | t_{su} | 5 | | | 2 |
| Input hold time from CLK | t_h | 0 | | | |

Note:

1. Minimum times are measured for 3.3V signalling environment according to the PCI Specification Rev. 2.1.
2. \overline{REQ} and \overline{GNT} are point-to-point signals. All other signals are bussed.

9.4.2 SPI Interface Timing

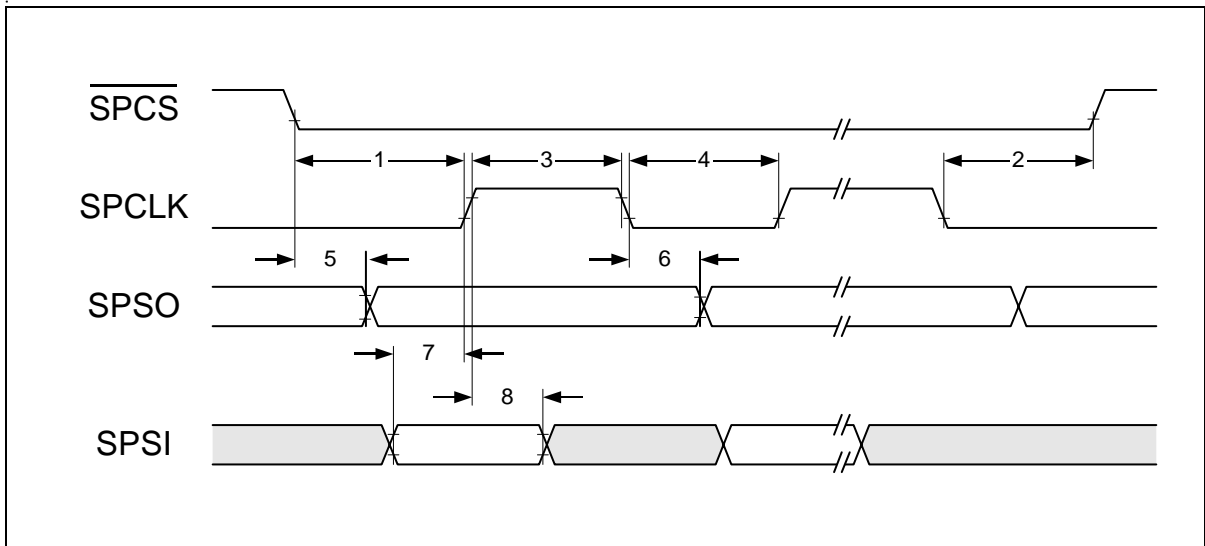


Figure 9-5 SPI Interface Timing

Table 9-7 SPI Interface Timing

| No. | Parameter | Limit Values | | Unit | Notes |
|-----|---|--------------|------|------|-------|
| | | min. | max. | | |
| 1 | $\overline{\text{SPCS}}$ low to SPCLK delay | 500 | | ns | 1 |
| 2 | SPCLK to $\overline{\text{SPCS}}$ delay | 500 | | ns | |
| 3 | SPCLK high time | 500 | | ns | |
| 4 | SPCLK low time | 500 | | ns | |
| 5 | $\overline{\text{SPCS}}$ to SPSO delay | | 100 | ns | |
| 6 | SPCLK to SPSO delay | | 100 | ns | |
| 7 | SPSI to SPCLK setup time | 100 | | ns | |
| 8 | SPSI to SPCLK hold time | 100 | | ns | |

Note:

- 1 SPI clock is related to PCI clock where the SPI frequency is 1/78 of the PCI frequency. All timings for SPI interface are calculated with a PCI clock running at 33 MHz.

9.4.3 Local Microprocessor Interface Timing

9.4.3.1 Intel Bus Interface Timing (Slave Mode)

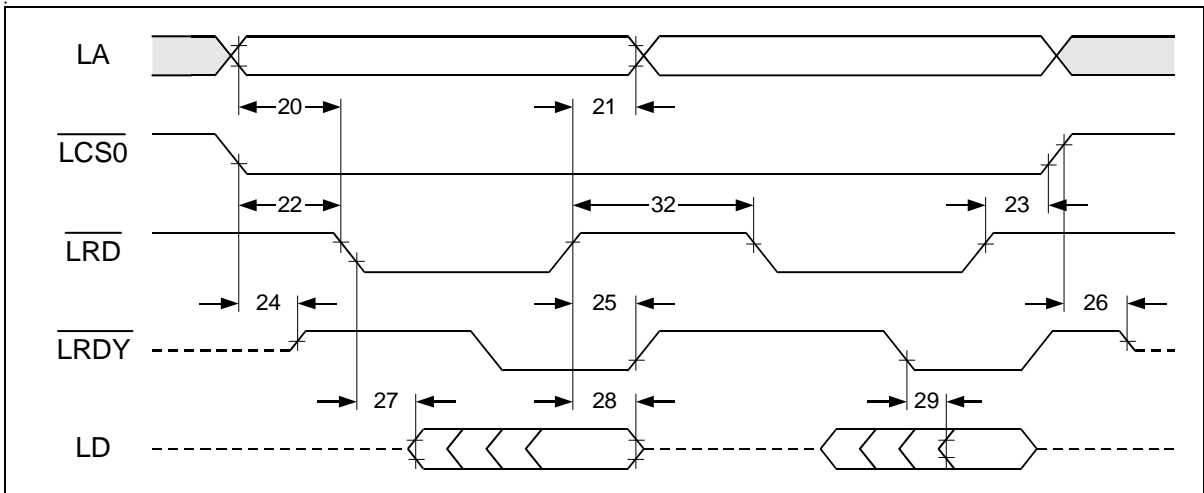


Figure 9-6 Intel Read Cycle Timing (Slave Mode)

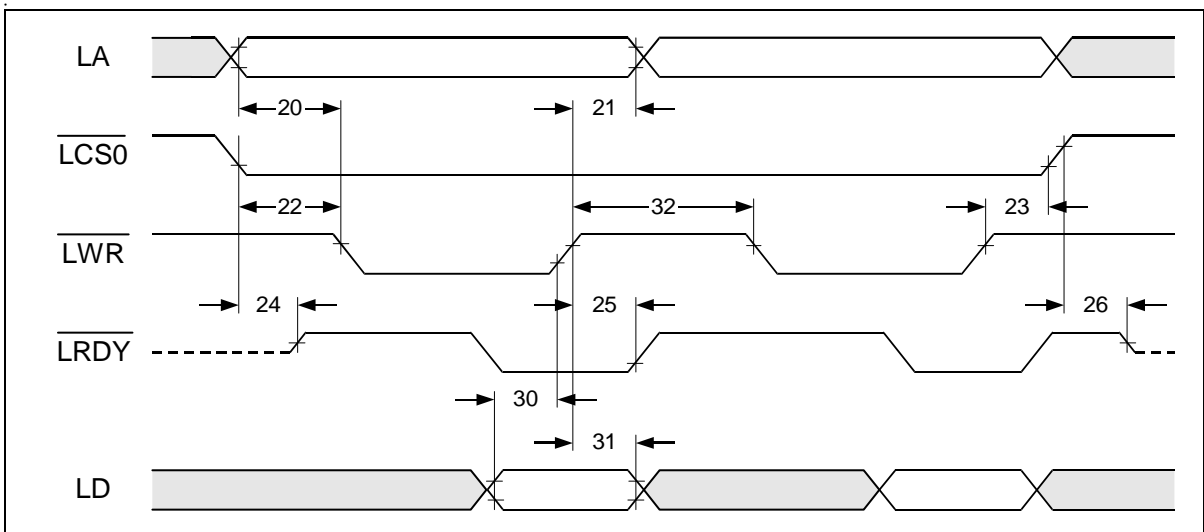


Figure 9-7 Intel Write Cycle Timing (Slave Mode)

Table 9-8 Intel Bus Interface Timing

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------|
| | | min. | max. | |
| 20 | LA to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ setup time | 20 | | ns |
| 21 | LA to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ hold time | 0 | | ns |
| 22 | $\overline{\text{LCS0}}$ to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ setup time | 20 | | ns |
| 23 | $\overline{\text{LCS0}}$ to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ hold time | 0 | | ns |
| 24 | $\overline{\text{LCS0}}$ low to $\overline{\text{LRDY}}$ active delay | | 20 | ns |
| 25 | $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ high to $\overline{\text{LRDY}}$ high delay | | 20 | ns |
| 26 | $\overline{\text{LCS0}}$ high to $\overline{\text{LRDY}}$ float delay | | 20 | ns |
| 27 | $\overline{\text{LRD}}$ low to LD active delay | | 20 | ns |
| 28 | $\overline{\text{LRD}}$ high to LD float delay | | 20 | ns |
| 29 | $\overline{\text{LRDY}}$ low to LD valid delay | | 20 | ns |
| 30 | LD to $\overline{\text{LWR}}$ setup time | 20 | | ns |
| 31 | LD to $\overline{\text{LWR}}$ hold time | 5 | | ns |
| 32 | $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ minimum high time | 20 | | ns |

9.4.3.2 Intel Bus Interface Timing (Master Mode)

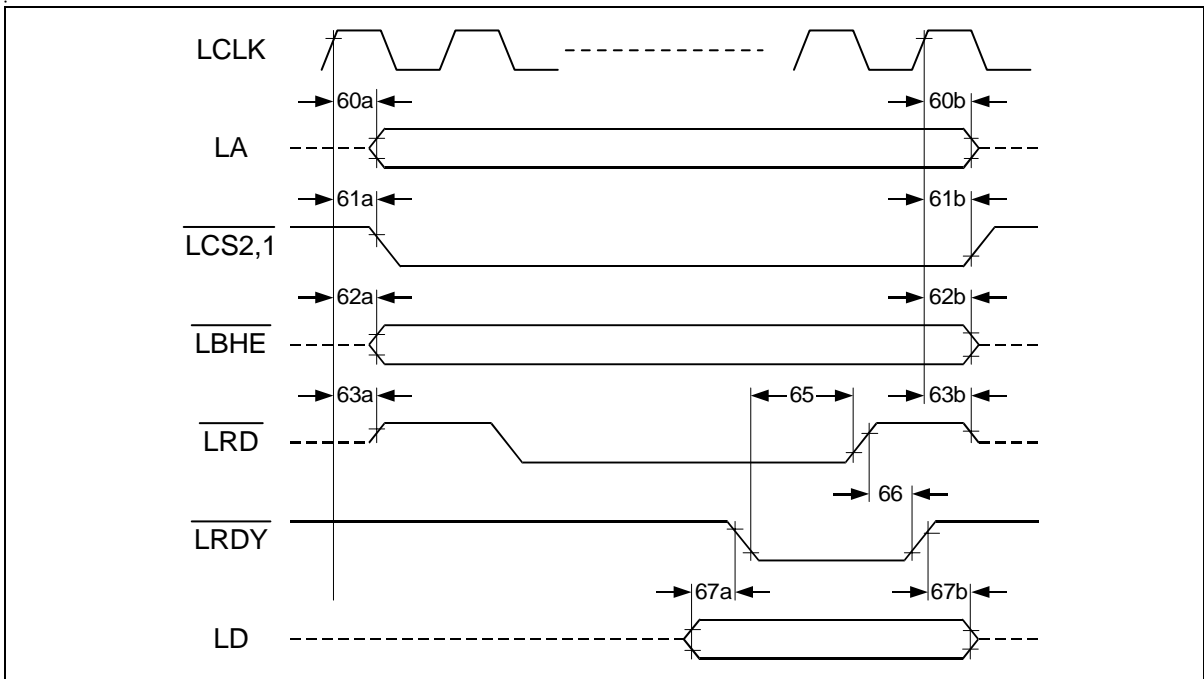


Figure 9-8 Intel Read Cycle Timing (Master Mode, $\overline{\text{LRDY}}$ controlled)

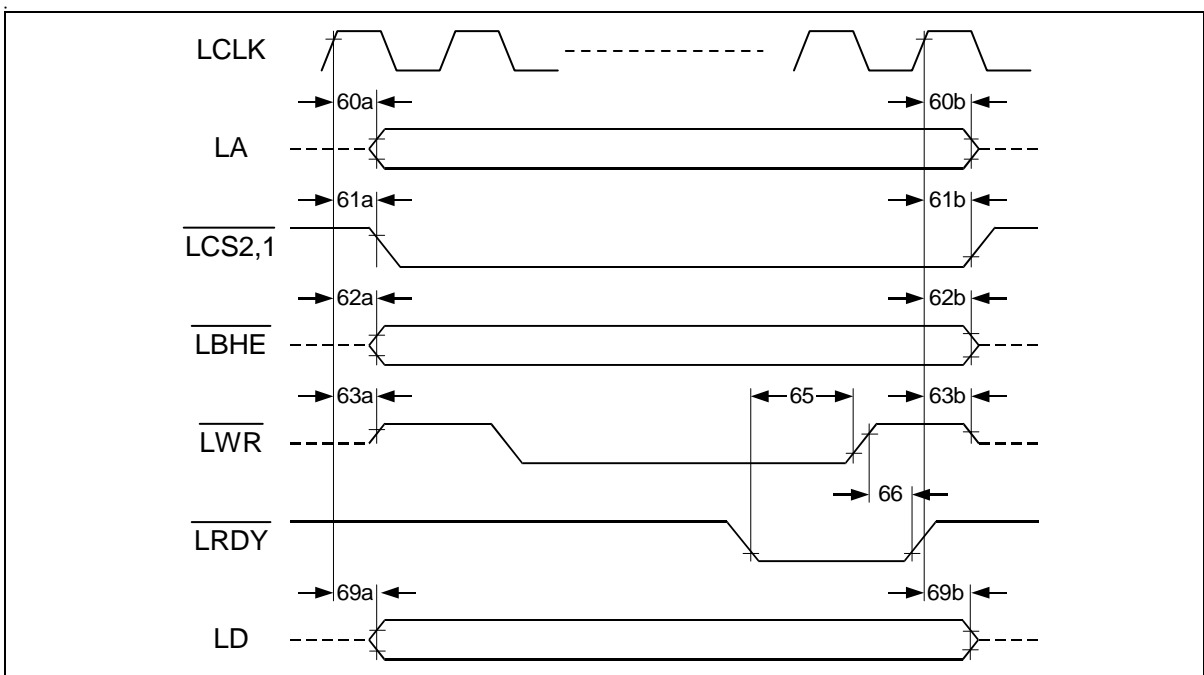


Figure 9-9 Intel Write Cycle Timing (Master Mode, $\overline{\text{LRDY}}$ controlled)

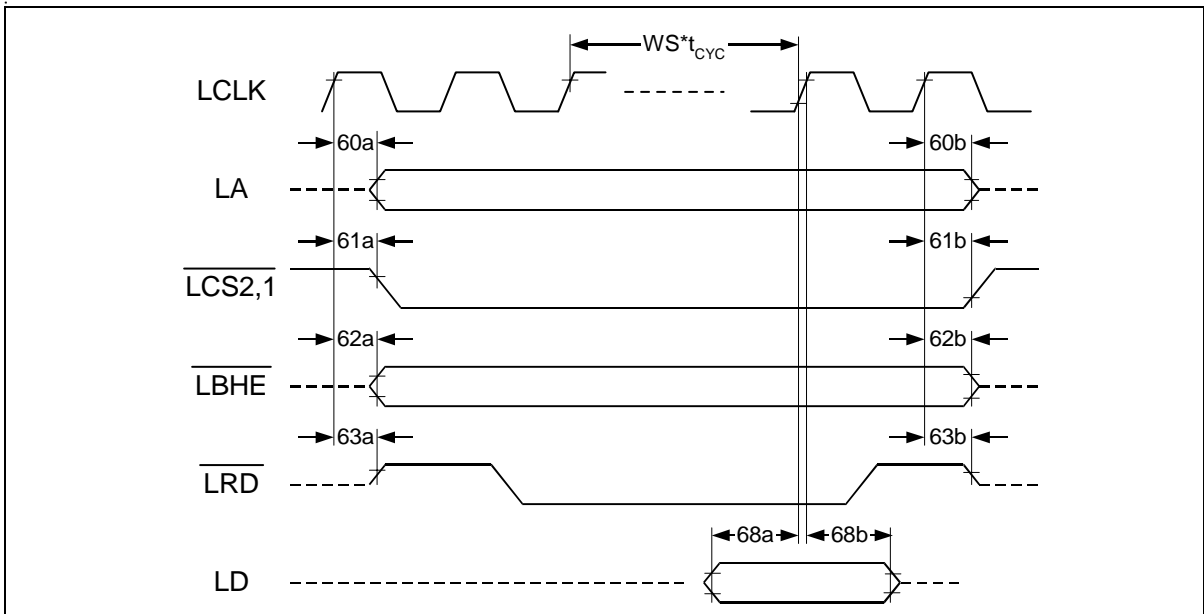


Figure 9-10 Intel Read Cycle Timing (Master Mode, Wait state controlled)

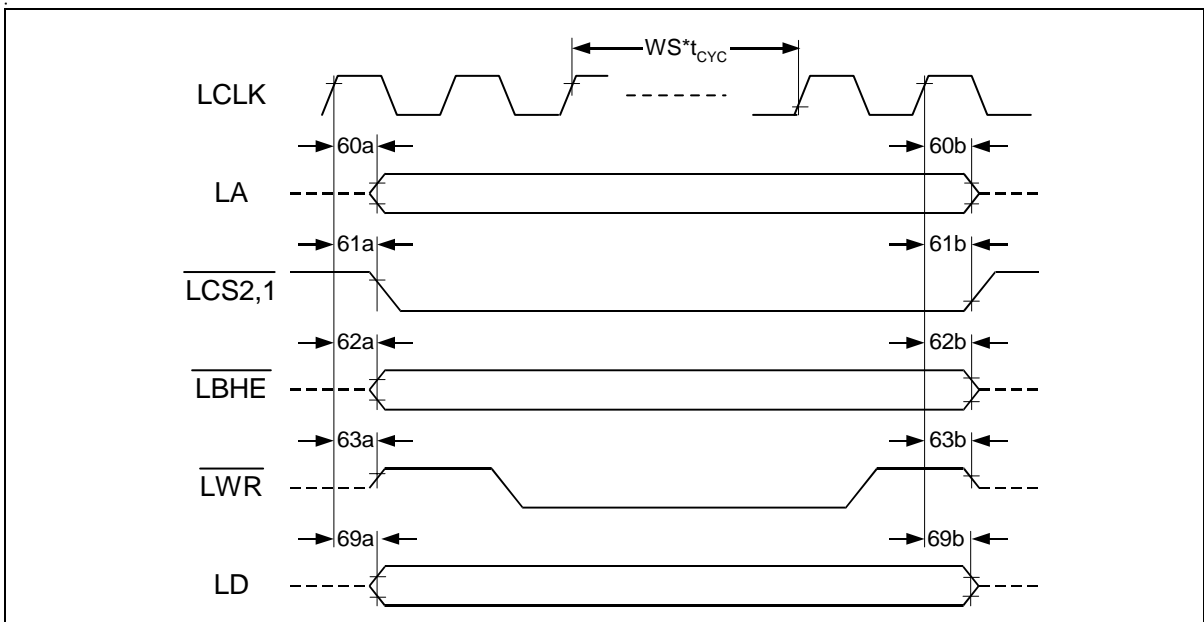


Figure 9-11 Intel Write Cycle Timing (Master Mode, Wait state controlled)

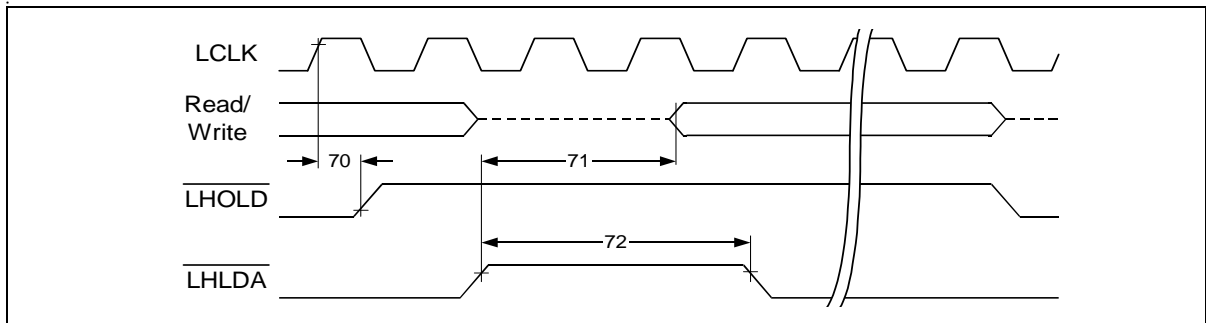


Figure 9-12 Intel Bus Arbitration Timing

Table 9-9 Intel Bus Interface Timing (Master Mode)

| No. | Parameter | Limit Values | | Unit |
|-----|--|--------------|------|------------------|
| | | min. | max. | |
| 60a | LCLK to LA active delay | 2 | 10 | ns |
| 60b | LCLK to LA float delay | 2 | 10 | ns |
| 61a | LCLK to $\overline{\text{LCS2,1}}$ active delay | 2 | 10 | ns |
| 61b | LCLK to $\overline{\text{LCS2,1}}$ float delay | 2 | 10 | ns |
| 62a | LCLK to $\overline{\text{LBHE}}$ active delay | 2 | 10 | ns |
| 62b | LCLK to $\overline{\text{LBHE}}$ float delay | 2 | 10 | ns |
| 63a | LCLK to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ active delay | 2 | 10 | ns |
| 63b | LCLK to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ float delay | 2 | 10 | ns |
| 65 | $\overline{\text{LRDY}}$ low to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ high delay | 2 | | t_{CYC} |
| 66 | $\overline{\text{LRDY}}$ to $\overline{\text{LRD}}$, $\overline{\text{LWR}}$ hold time | 0 | | ns |
| 67a | LD to $\overline{\text{LRDY}}$ setup time | 0 | | ns |
| 67b | LD to $\overline{\text{LRDY}}$ hold time | 0 | | ns |
| 68a | LD to LCLK setup time | 10 | | ns |
| 68b | LD to LCLK hold time | 5 | | ns |
| 69a | LCLK to LD delay | 2 | 10 | ns |
| 69b | LCLK to LD float delay | 2 | 10 | ns |
| 70 | LCLK to LHOLD delay | 2 | 10 | ns |
| 71 | LHLDA asserted to Read/Write Cycle start | 1 | | t_{CYC} |
| 72 | LHLDA minimum pulse width | 2 | | t_{CYC} |

Note: t_{CYC} is the clock period of the PCI clock.

9.4.3.3 Motorola Bus Interface Timing (Slave Mode)

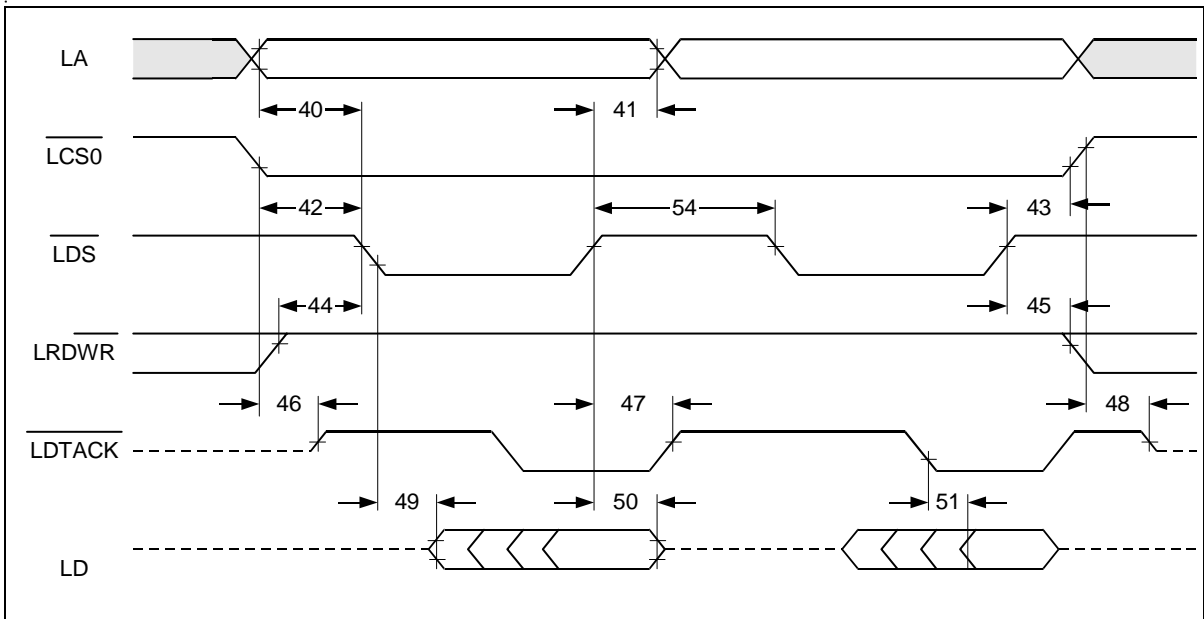


Figure 9-13 Motorola Read Cycle Timing (Slave Mode)

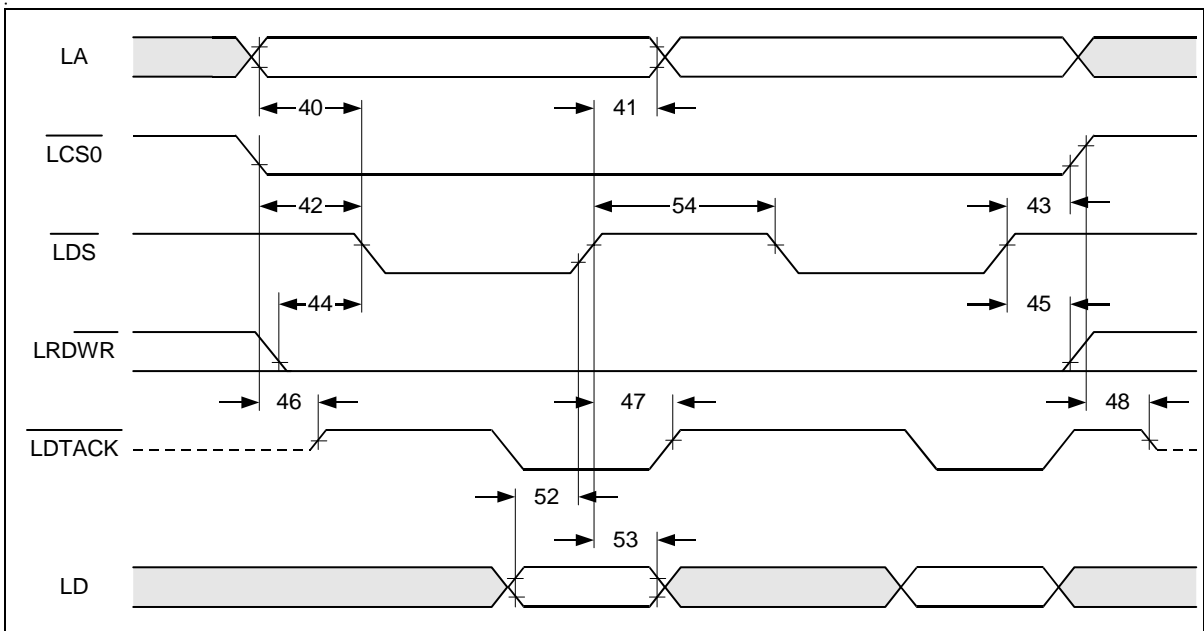


Figure 9-14 Motorola Write Cycle Timing (Slave Mode)

Table 9-10 Motorola Bus Interface Timing

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------|
| | | min. | max. | |
| 40 | LA to $\overline{\text{LDS}}$ setup time | 20 | | ns |
| 41 | LA to $\overline{\text{LDS}}$ hold time | 0 | | ns |
| 42 | $\overline{\text{LCS0}}$ to $\overline{\text{LDS}}$ setup time | 20 | | ns |
| 43 | $\overline{\text{LCS0}}$ to $\overline{\text{LDS}}$ hold time | 0 | | ns |
| 44 | LRDWR to $\overline{\text{LDS}}$ setup time | 20 | | ns |
| 45 | LRDWR to $\overline{\text{LDS}}$ hold time | 0 | | ns |
| 46 | $\overline{\text{LCS0}}$ low to $\overline{\text{LDTACK}}$ active delay | | 20 | ns |
| 47 | $\overline{\text{LDS}}$ high to $\overline{\text{LDTACK}}$ high delay | | 20 | ns |
| 48 | $\overline{\text{LCS0}}$ high to $\overline{\text{LDTACK}}$ float delay | | 20 | ns |
| 49 | $\overline{\text{LDS}}$ low to LD active delay | | 20 | ns |
| 50 | $\overline{\text{LDS}}$ high to LD float delay | | 20 | ns |
| 51 | $\overline{\text{LDTACK}}$ low to LD valid delay | | 20 | ns |
| 52 | LD to $\overline{\text{LDS}}$ setup time | 20 | | ns |
| 53 | LD to $\overline{\text{LDS}}$ hold time | 5 | | ns |
| 54 | $\overline{\text{LDS}}$ minimum high time | 20 | | ns |

9.4.3.4 Motorola Bus Interface Timing (Master Mode)

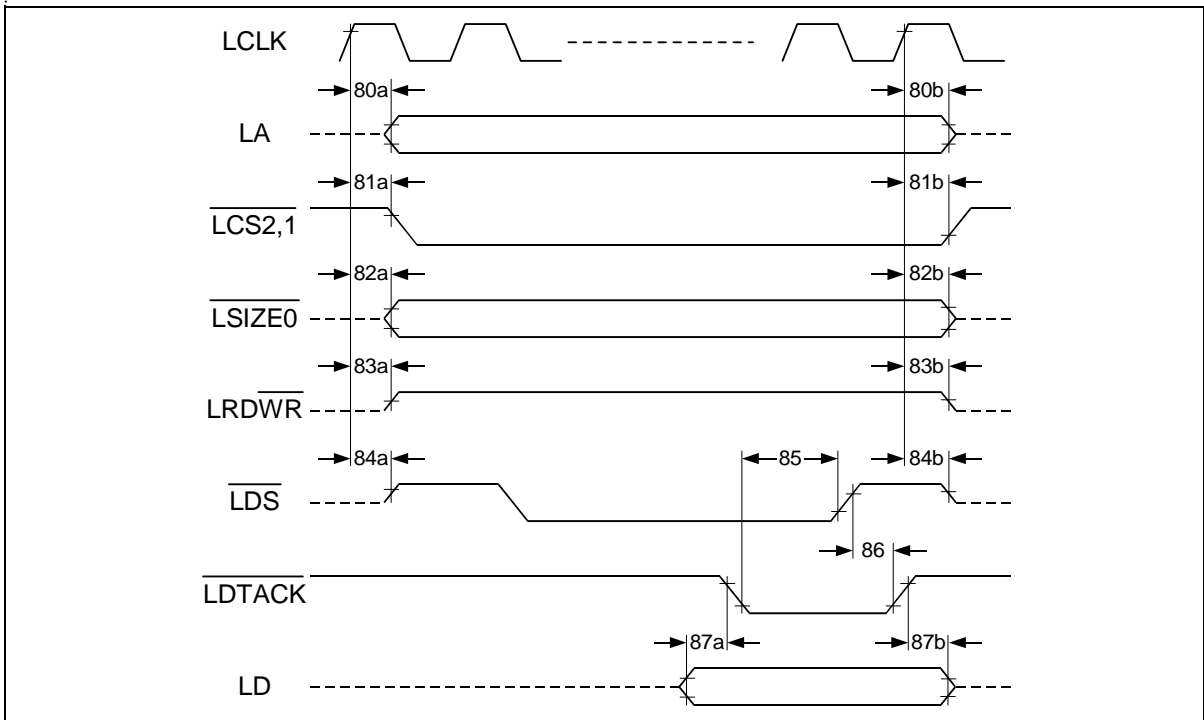


Figure 9-15 Motorola Read Cycle Timing (Master Mode, LDTACK controlled)

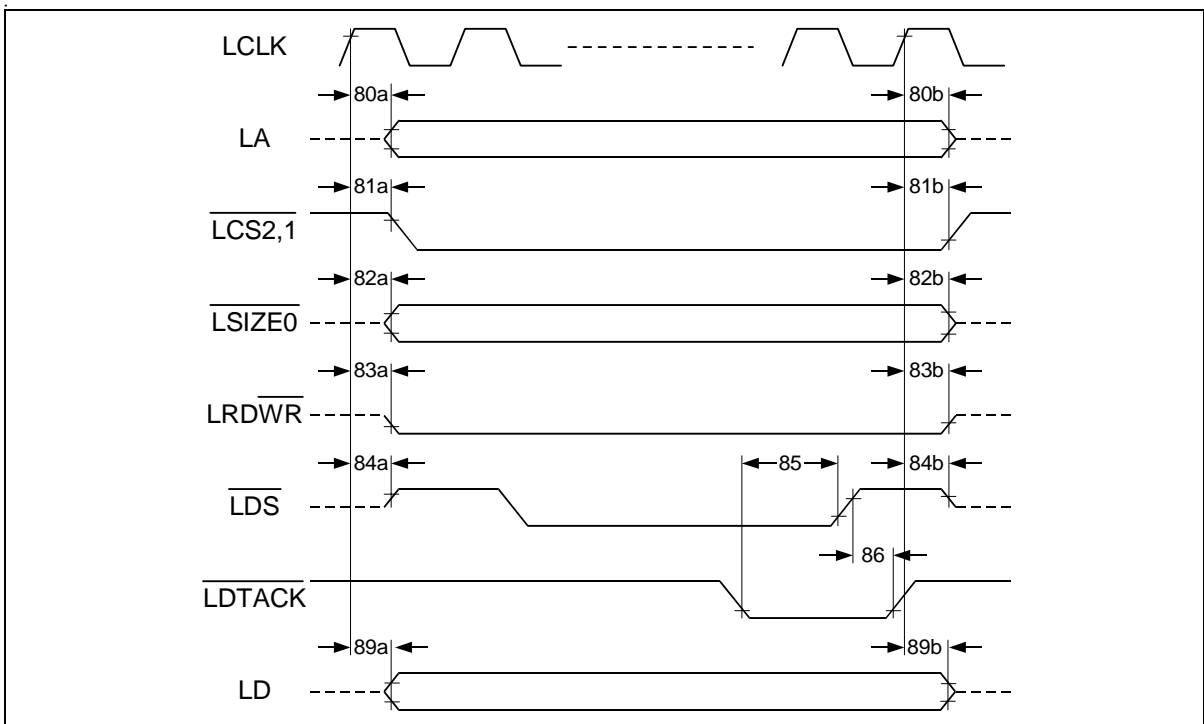


Figure 9-16 Motorola Write Cycle Timing (Master Mode, LDTACK controlled)

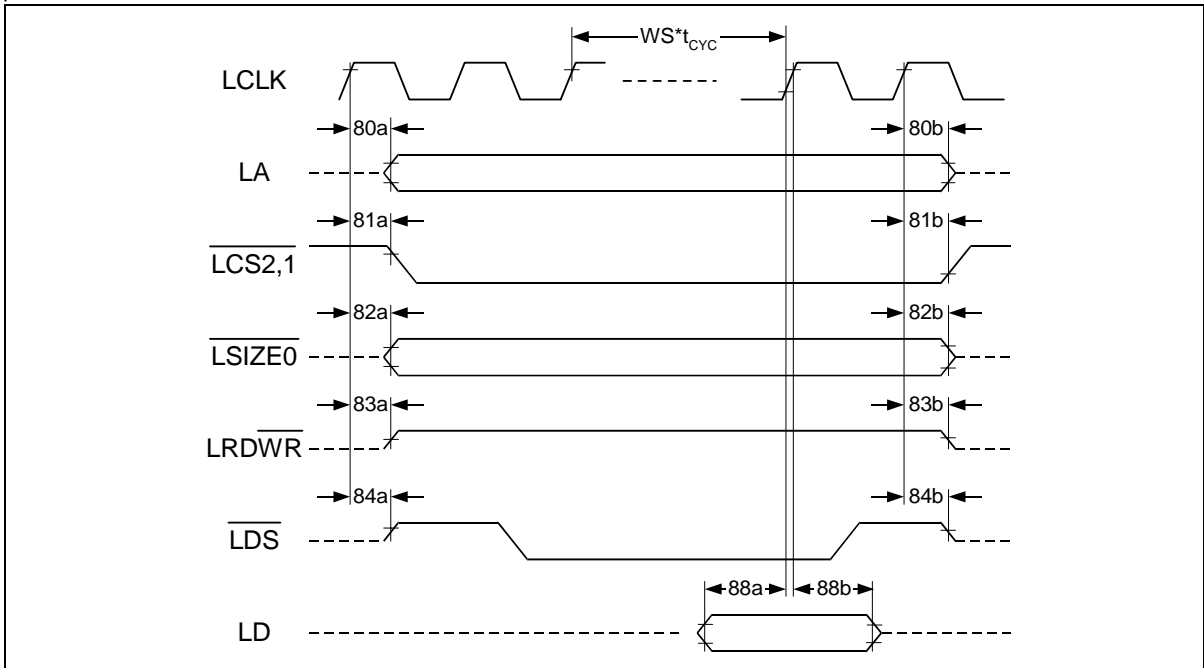


Figure 9-17 Motorola Read Cycle Timing (Master Mode, Wait state controlled)

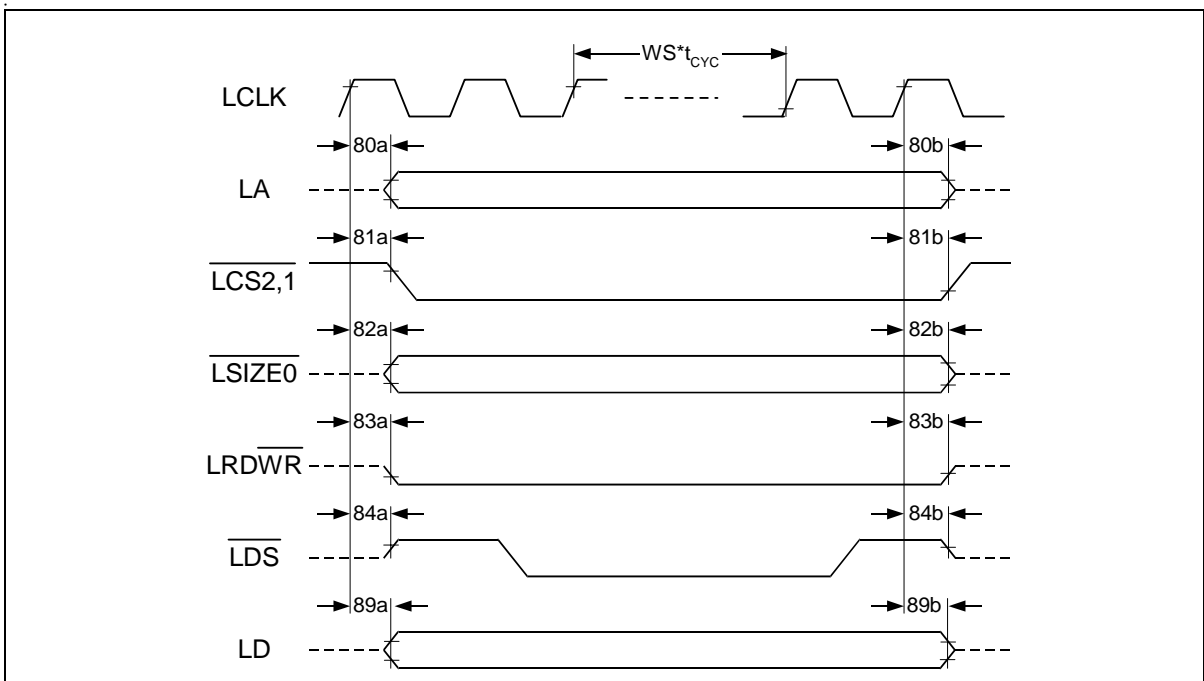


Figure 9-18 Motorola Write Cycle Timing (Master Mode, Wait state controlled)

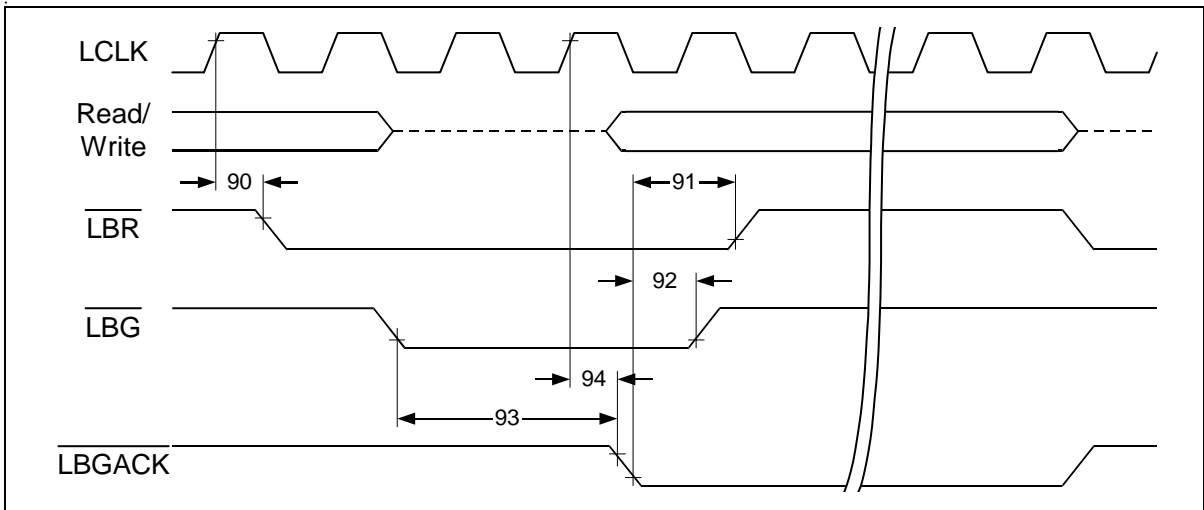


Figure 9-19 Motorola Bus Arbitration Timing

Table 9-11 Motorola Bus Interface Timing (Master Mode)

| No. | Parameter | Limit Values | | Unit |
|-----|--|--------------|------|------------------|
| | | min. | max. | |
| 80a | LCLK to LA active delay | 2 | 10 | ns |
| 80b | LCLK to LA float delay | 2 | 10 | ns |
| 81a | LCLK to $\overline{\text{LCS2,1}}$ active delay | 2 | 10 | ns |
| 81b | LCLK to $\overline{\text{LCS2,1}}$ float delay | 2 | 10 | ns |
| 82a | LCLK to LSIZE0 active delay | 2 | 10 | ns |
| 82b | LCLK to LSIZE0 float delay | 2 | 10 | ns |
| 83a | LCLK to $\overline{\text{LRDWR}}$ active delay | 2 | 10 | ns |
| 83b | LCLK to $\overline{\text{LRDWR}}$ float delay | 2 | 10 | ns |
| 84a | LCLK to $\overline{\text{LDS}}$ active delay | 2 | 10 | ns |
| 84b | LCLK to $\overline{\text{LDS}}$ float delay | 2 | 10 | ns |
| 85 | $\overline{\text{LDTACK}}$ low to $\overline{\text{LDS}}$ high delay | 2 | | t_{CYC} |
| 86 | $\overline{\text{LDTACK}}$ to $\overline{\text{LDS}}$ hold time | 0 | | ns |
| 87a | LD to $\overline{\text{LDTACK}}$ setup time | 0 | | ns |
| 87b | LD to $\overline{\text{LDTACK}}$ hold time | 0 | | ns |
| 88a | LD to LCLK setup time | 10 | | ns |
| 88b | LD to LCLK hold time | 5 | | ns |
| 89a | LCLK to LD delay | 2 | 10 | ns |

Electrical Characteristics

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------------------|
| | | min. | max. | |
| 89b | LCLK to LD float delay | 2 | 10 | ns |
| 90 | LCLK to $\overline{\text{LBR}}$ delay | 2 | 10 | ns |
| 91 | $\overline{\text{LBGACK}}$ to $\overline{\text{LBR}}$ delay | 1 | | t_{CYC} |
| 92 | $\overline{\text{LBG}}$ to $\overline{\text{LBGACK}}$ hold time | 0 | | ns |
| 93 | $\overline{\text{LBG}}$ to $\overline{\text{LBGACK}}$ delay | 1 | | t_{CYC} |
| 94 | LCLK to $\overline{\text{LBGACK}}$ delay | 2 | 10 | ns |

Note: t_{CYC} is the clock period of the PCI clock.

9.4.4 Serial Interface Timing

9.4.4.1 DS3 Serial Interface Timing

Note: The clock input timings are calculated assuming a PCI clock frequency of 33 MHz or more.

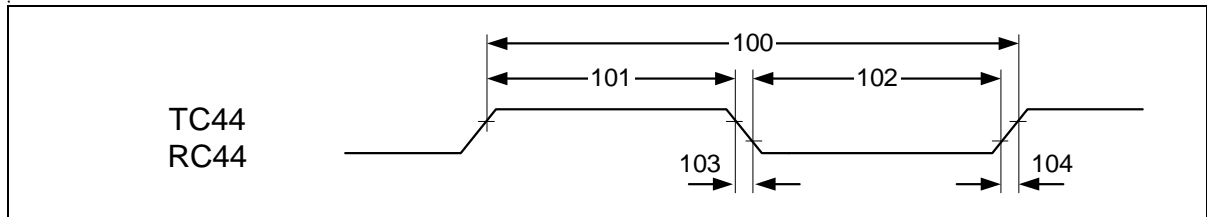


Figure 9-20 Clock Input Timing

Table 9-12 Clock Input Timing

| No. | Parameter | Limit Values | | Unit |
|-----|-------------------|--------------|------|------|
| | | min. | max. | |
| 100 | Clock frequency | nom. 44.736 | | MHz |
| 101 | Clock high timing | 7.5 | | ns |
| 102 | Clock low timing | 7.5 | | ns |
| 103 | Clock fall time | | 2 | ns |
| 104 | Clock rise time | | 2 | ns |

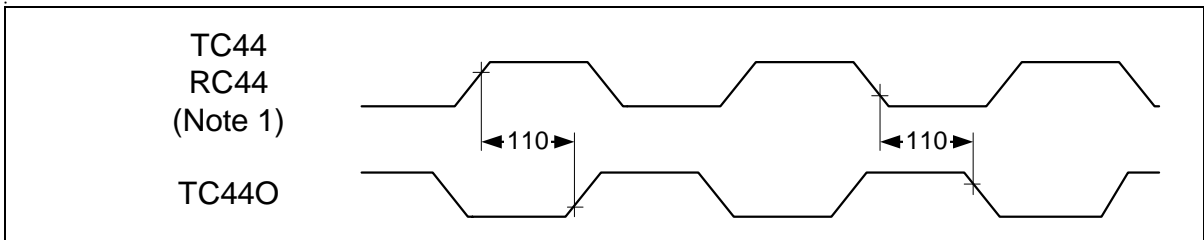


Figure 9-21 DS3 Transmit Cycle Timing

Note:

1. Actual clock reference depends on selected clock mode:

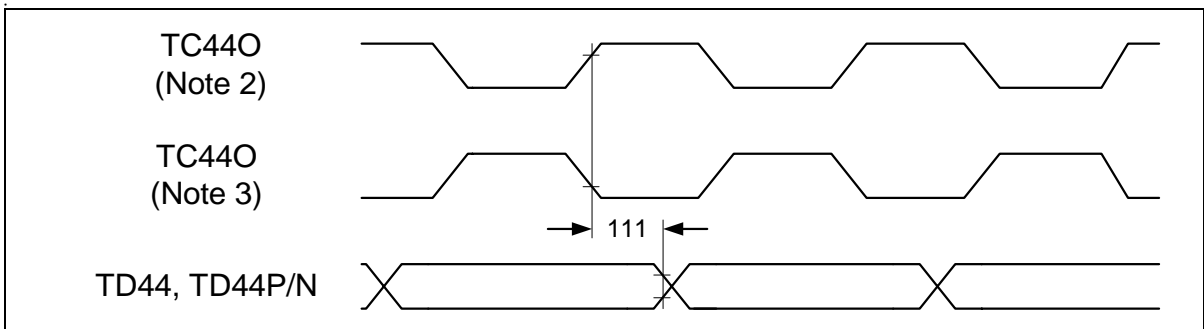


Figure 9-22 DS3 Transmit Data Timing

Note:

2. Timing for transmit data which is updated on the rising edge of TC44O.
3. Timing for transmit data which is updated on the falling edge of TC44O.

Table 9-13 DS3 Transmit Cycle Timing

| No. | Parameter | Limit Values | | Unit |
|-----|----------------------------------|--------------|------|------|
| | | min. | max. | |
| 110 | RC44, TC44 to TC44O delay | 2 | 15 | ns |
| 111 | TC44O to TD44, TD44P/TD44N delay | -5 | 5 | ns |

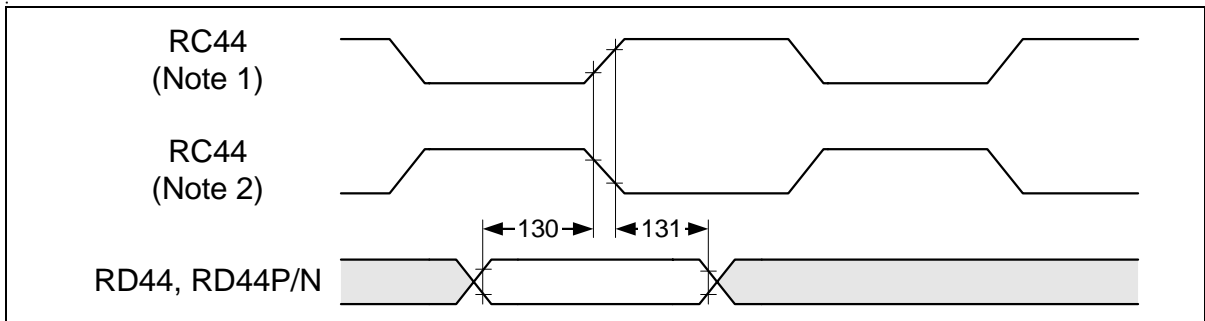


Figure 9-23 DS3 Receive Cycle Timing

Note:

1. *Timing for data which is sampled on the rising edge of the receive clock.*
2. *Timing for data which is sampled on the falling edge of the receive clock.*

Table 9-14 DS3 Receive Cycle Timing

| No. | Parameter | Limit Values | | Unit |
|-----|--------------------------------------|--------------|------|------|
| | | min. | max. | |
| 130 | RD44, RD44P/RD44N to RC44 setup time | 5 | | ns |
| 131 | RD44, RD44P/RD44N to RC44 hold time | 5 | | ns |

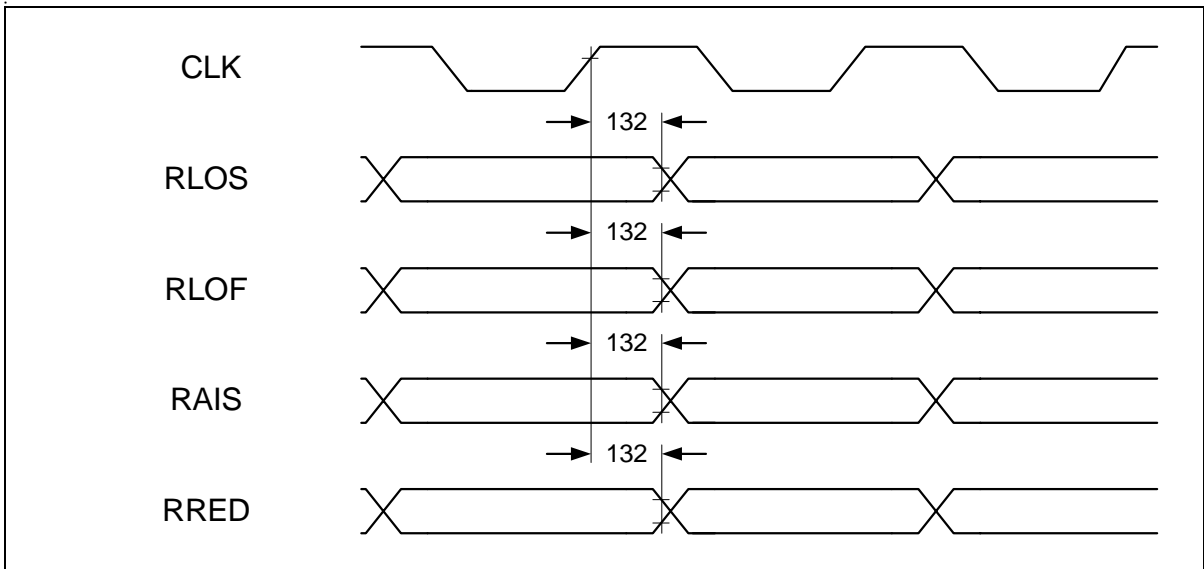


Figure 9-24 DS3 Status Signal Timing

Note: Status signals are generated synchronous to the PCI clock.

Table 9-15 DS3 Status Signal Timing

| No. | Parameter | Limit Values | | Unit |
|-----|----------------------------------|--------------|------|------|
| | | min. | max. | |
| 132 | CLK to RLOS/RLOF/RAIS/RRED delay | 2 | 10 | ns |

9.4.4.2 Overhead Bit Timing

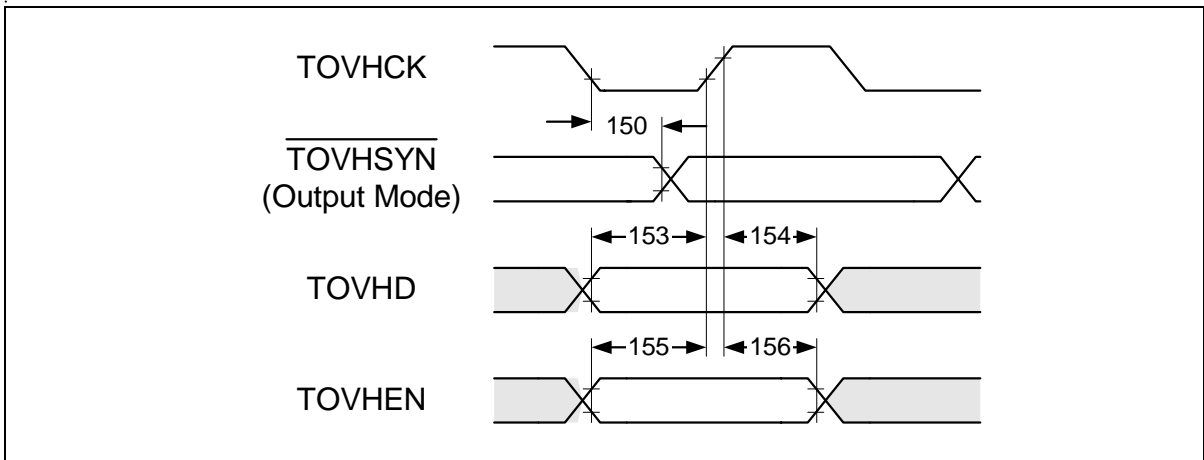


Figure 9-25 DS3 Transmit Overhead Timing

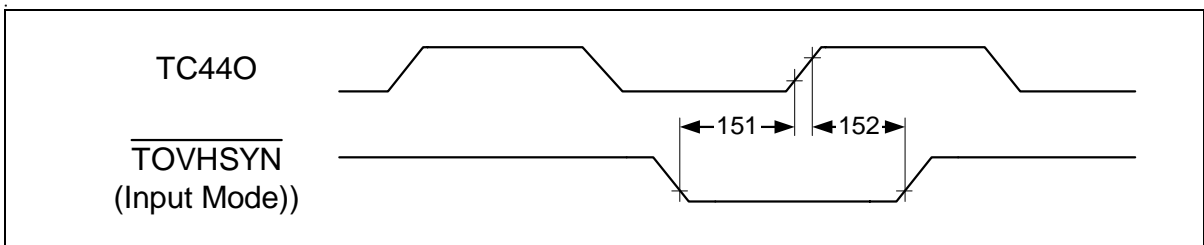


Figure 9-26 DS3 Transmit Overhead Synchronization Timing

Table 9-16 DS3 Transmit Overhead Timing

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------|
| | | min. | max. | |
| 150 | $\overline{\text{TOVHCK}}$ to $\overline{\text{TOVHSYN}}$ delay | | 75 | ns |
| 151 | $\overline{\text{TOVHSYN}}$ to TCLKO44 setup time | 5 | | ns |
| 152 | $\overline{\text{TOVHSYN}}$ to TCLKO44 hold time | 5 | | ns |
| 153 | TOVD to TOVHCK setup time | 25 | | ns |
| 154 | TOVD to TOVHCK hold time | 5 | | ns |
| 155 | TOVHEN to TOVHCK setup time | 25 | | ns |
| 156 | TOVHEN to TOVHCK hold time | 5 | | ns |

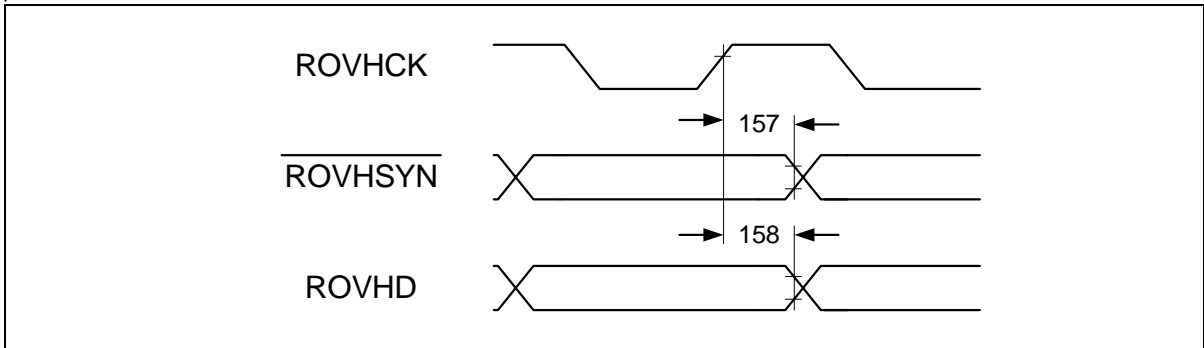


Figure 9-27 DS3 Receive Overhead Timing

Table 9-17 DS3 Receive Overhead Timing

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------|
| | | min. | max. | |
| 157 | ROVHCK to $\overline{\text{ROVHSYN}}$ delay | | 75 | ns |
| 158 | ROVHCK to ROVHD delay | | 75 | ns |

9.4.4.3 Stuff Bit Timing

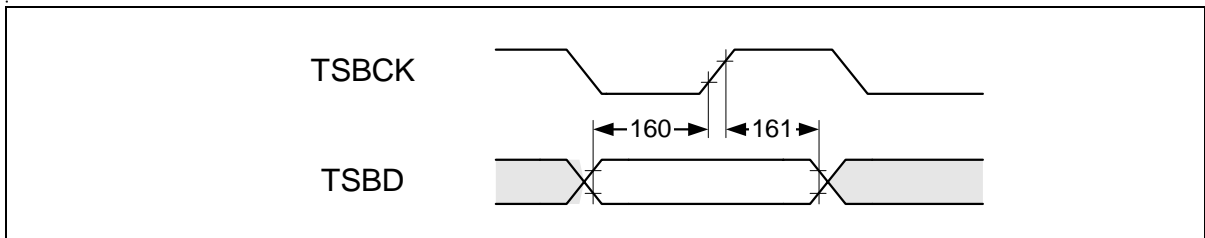


Figure 9-28 DS3 Transmit Stuff Bit Timing

Table 9-18 DS3 Transmit Stuff Timing

| No. | Parameter | Limit Values | | Unit |
|-----|--------------------------|--------------|------|------|
| | | min. | max. | |
| 160 | TSBD to TSBCK setup time | 25 | | ns |
| 161 | TSBD to TSBCK hold time | 5 | | ns |

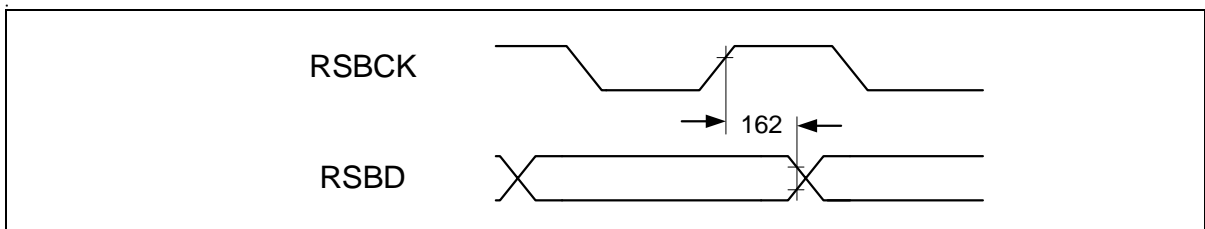


Figure 9-29 DS3 Receive Stuff Bit Timing

Table 9-19 DS3 Receive Stuff Bit Timing

| No. | Parameter | Limit Values | | Unit |
|-----|--------------------|--------------|------|------|
| | | min. | max. | |
| 162 | RSBCK to RSD delay | | 75 | ns |

9.4.4.4 T1/E1 Tributary Timing

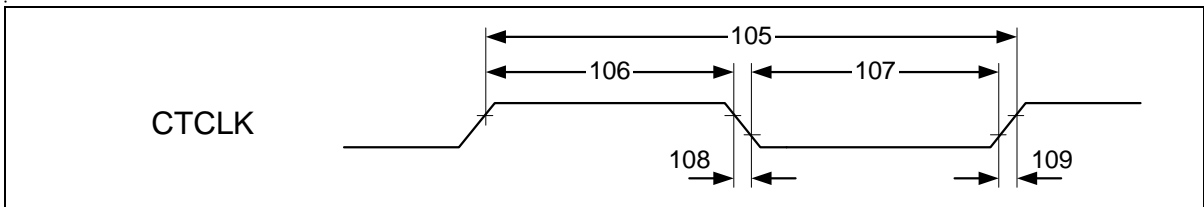


Figure 9-30 T1/E1 Tributary Clock Input Timing

Table 9-20 T1/E1 Tributary Clock Input Timing

| No. | Parameter | Limit Values | | | Unit |
|--|-------------------|-------------------------|-----|------|------|
| | | min. | typ | max. | |
| Tributaries operated in E1 Mode | | | | | |
| 105 | Clock frequency | 2.048 MHz \pm 50 ppm | | | |
| 106 | Clock high timing | 40 | | | ns |
| 107 | Clock low timing | 40 | | | ns |
| 108 | Clock fall time | | | 10 | ns |
| 109 | Clock rise time | | | 10 | ns |
| Tributaries operated in T1 Mode | | | | | |
| 105 | Clock frequency | 1.544 MHz \pm 130 ppm | | | |
| 106 | Clock high timing | 40 | | | ns |
| 107 | Clock low timing | 40 | | | ns |
| 108 | Clock fall time | | | 10 | ns |
| 109 | Clock rise time | | | 10 | ns |

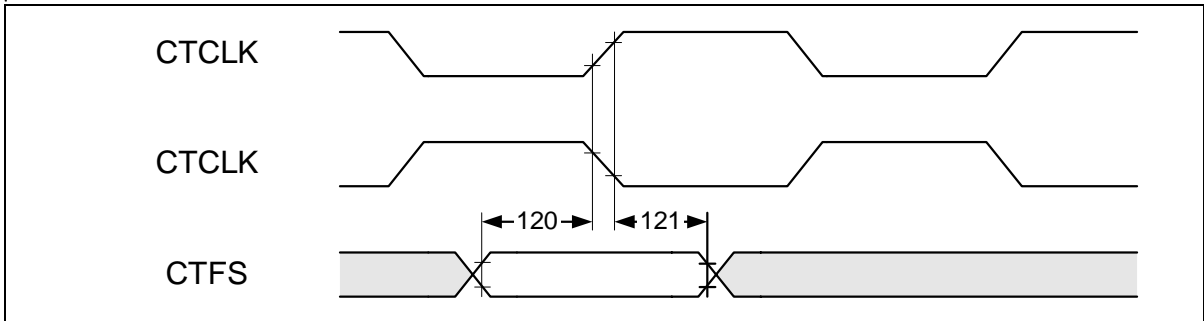


Figure 9-31 T1/E1 Tributary Synchronization Timing

Table 9-21 T1/E1 Tributary Synchronization Timing

| No. | Parameter | Limit Values | | Unit |
|-----|--------------------------|--------------|------|------|
| | | min. | max. | |
| 120 | CTFS to CTCLK setup time | 5 | | ns |
| 121 | CTFS to CTCLK hold time | 5 | | ns |

9.4.4.5 Test Port Timing

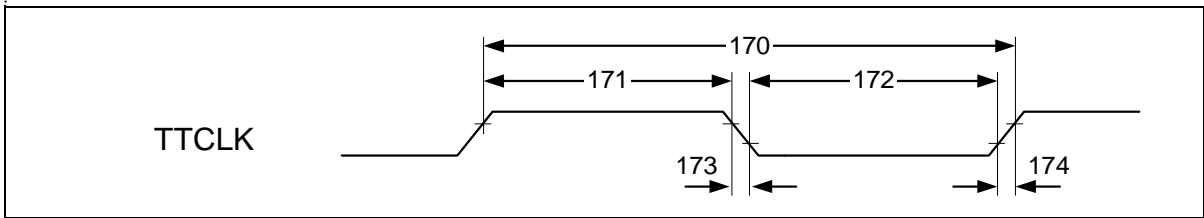


Figure 9-32 T1/E1 Test Transmit Clock Timing

Table 9-22 T1/E1 Test Transmit Clock Timing

| No. | Parameter | Limit Values | | | Unit |
|--------------------------------------|-------------------|-------------------------|-----|------|------|
| | | min. | typ | max. | |
| Test port operated in E1 Mode | | | | | |
| 170 | Clock period | 2.048 MHz \pm 50 ppm | | | |
| 171 | Clock high timing | 100 | | | ns |
| 172 | Clock low timing | 100 | | | ns |
| 173 | Clock fall time | | | 10 | ns |
| 174 | Clock rise time | | | 10 | ns |
| Test port operated in T1 Mode | | | | | |
| 170 | Clock period | 1.544 MHz \pm 130 ppm | | | |
| 171 | Clock high timing | 100 | | | ns |
| 172 | Clock low timing | 100 | | | ns |
| 173 | Clock fall time | | | 10 | ns |
| 174 | Clock rise time | | | 10 | ns |

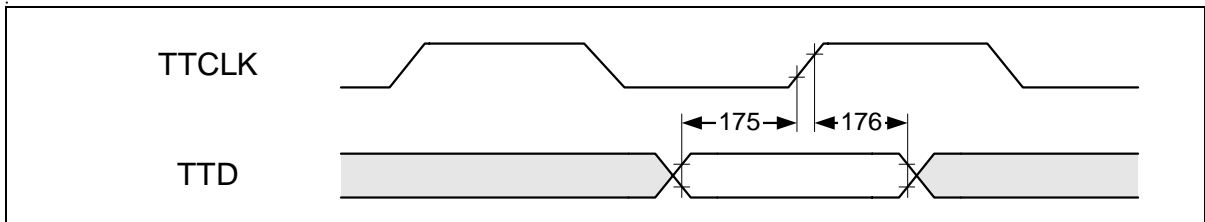


Figure 9-33 T1/E1 Test Transmit Data Timing

Table 9-23 T1/E1 Test Transmit Data Timing

| No. | Parameter | Limit Values | | Unit |
|-----|-----------------------------|--------------|------|------|
| | | min. | max. | |
| 175 | TTD(x) to TTC(x) setup time | 25 | | ns |
| 176 | TTD(x) to TTC(x) hold time | 75 | | ns |

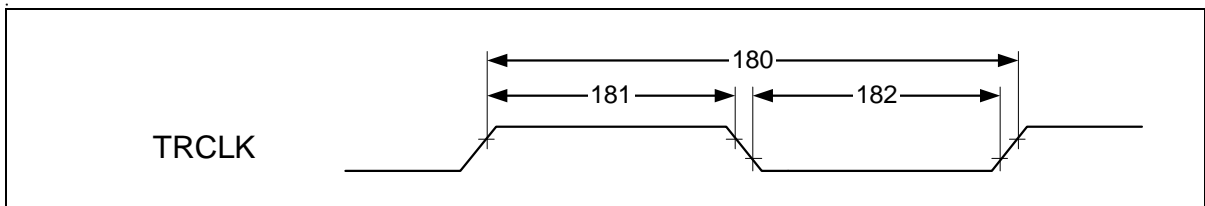


Figure 9-34 T1/E1 Test Receive Clock Timing

Table 9-24 T1/E1 Test Receive Clock Timing

| No. | Parameter | Limit Values | | | Unit |
|--------------------------------------|-------------------|--------------|-----|------|------|
| | | min. | typ | max. | |
| Test port operated in E1 Mode | | | | | |
| 180 | Clock period | 469 | | 2056 | ns |
| 181 | Clock high timing | 156 | | 335 | ns |
| 182 | Clock low timing | 312 | | 1900 | ns |
| Test Port operated in T1 Mode | | | | | |
| 180 | Clock period | 625 | | 1587 | ns |
| 181 | Clock high timing | 310 | | 495 | ns |
| 182 | Clock low timing | 310 | | 1275 | ns |

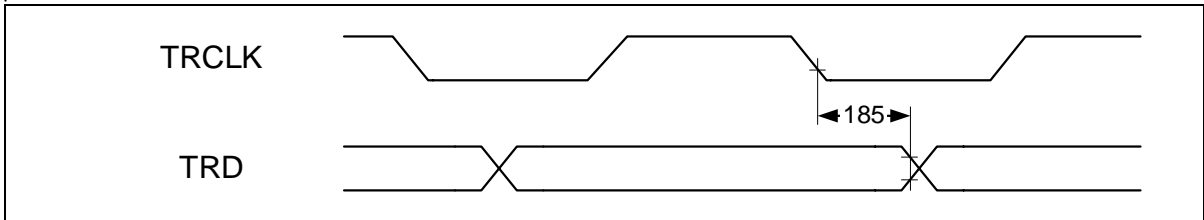


Figure 9-35 T1/E1 Test Receive Data Timing

Table 9-25 Test T1/E1 Receive Data Timing

| No. | Parameter | Limit Values | | Unit |
|-----|------------------------|--------------|------|------|
| | | min. | max. | |
| 185 | RTC(x) to RTD(x) delay | -5 | 25 | ns |

9.4.5 JTAG Interface Timing

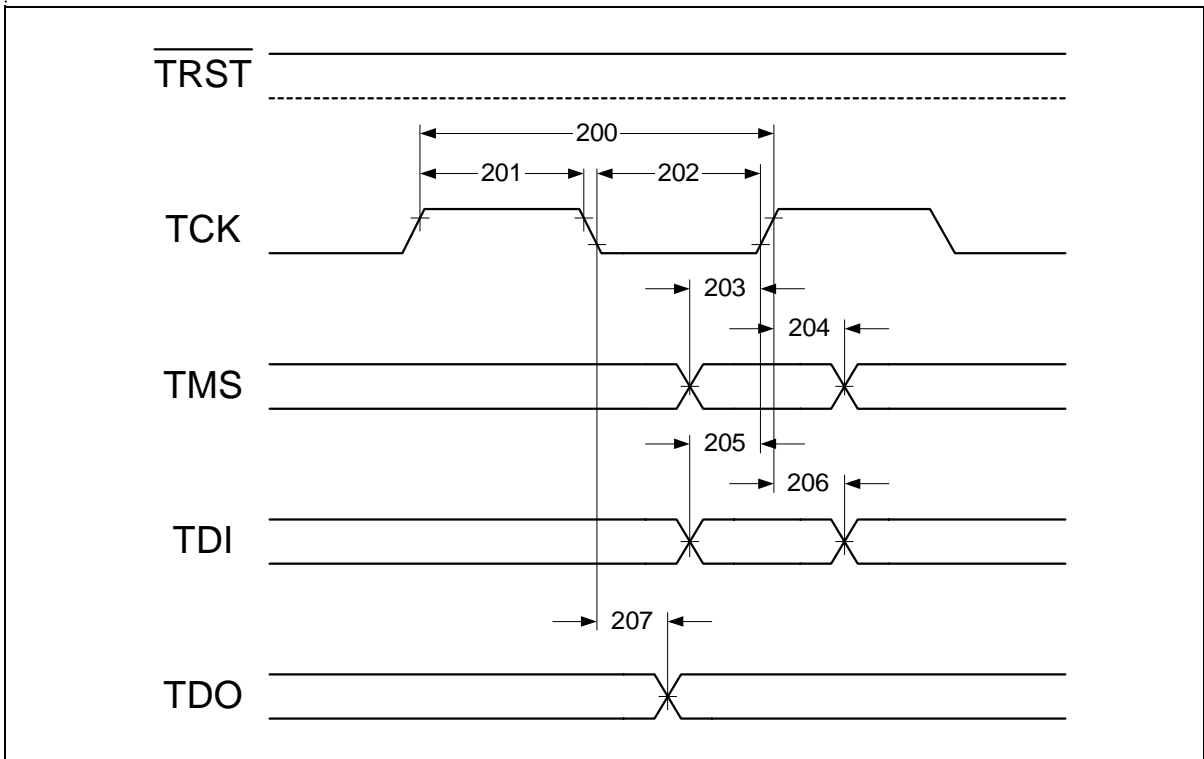


Figure 9-36 JTAG Interface Timing

Table 9-26 JTAG Interface Timing

| No. | Parameter | Limit Values | | Unit |
|-----|----------------|--------------|------|------|
| | | min. | max. | |
| 200 | TCK period | 120 | | ns |
| 201 | TCK high time | 60 | | ns |
| 202 | TCK low time | 60 | | ns |
| 203 | TMS setup time | 20 | | ns |
| 204 | TMS hold time | 20 | | ns |
| 205 | TDI setup time | 20 | | ns |
| 206 | TDI hold time | 20 | | ns |
| 207 | TDO valid time | 50 | | ns |

9.4.6 Reset Timing

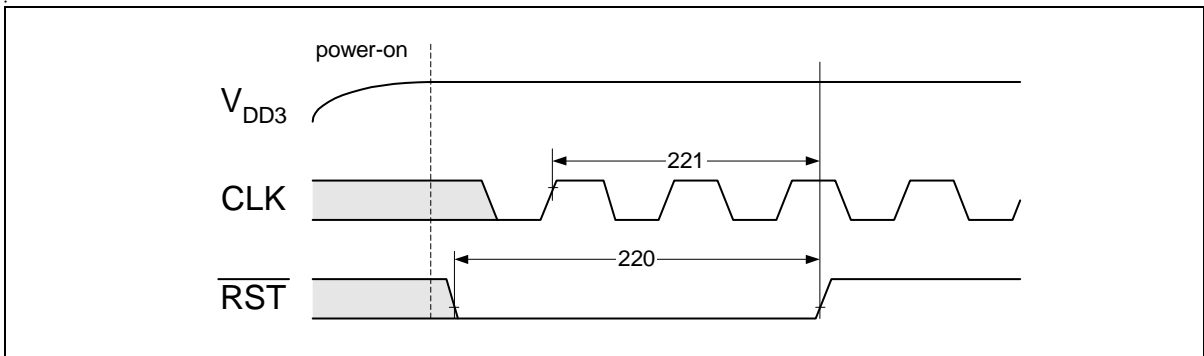


Figure 9-37 Reset Timing

Table 9-27 Reset Timing

| No. | Parameter | Limit Values | | Unit |
|-----|---|--------------|------|------------|
| | | min. | max. | |
| 220 | \overline{RST} pulse width | 120 | | ns |
| 221 | Number of CLK cycles during \overline{RST} active | 2 | | CLK cycles |

10 Package Outline

