

---

# TPMC863

## 4 Channel High Speed Synch/Asynch Serial Interface

Version 1.0

### User Manual

Issue 1.1

March 2007

---

**TEWS TECHNOLOGIES GmbH**

Am Bahnhof 7  
25469 Halstenbek, Germany  
[www.tews.com](http://www.tews.com)

Phone: +49-(0)4101-4058-0  
Fax: +49-(0)4101-4058-19  
e-mail: [info@tews.com](mailto:info@tews.com)

**TEWS TECHNOLOGIES LLC**

9190 Double Diamond Parkway,  
Suite 127, Reno, NV 89521, USA  
[www.tews.com](http://www.tews.com)

Phone: +1 (775) 850 5830  
Fax: +1 (775) 201 0347  
e-mail: [usasales@tews.com](mailto:usasales@tews.com)

**TPMC863-10**

4 Channel High Speed Synch/Asynch Serial Interface

HD68 front panel connector

P14 Back I/O

This document contains information, which is proprietary to TEWS TECHNOLOGIES GmbH. Any reproduction without written permission is forbidden.

TEWS TECHNOLOGIES GmbH has made any effort to ensure that this manual is accurate and complete. However TEWS TECHNOLOGIES GmbH reserves the right to change the product described in this document at any time without notice.

TEWS TECHNOLOGIES GmbH is not liable for any damage arising out of the application or use of the device described herein.

**Style Conventions**

Hexadecimal characters are specified with prefix 0x, i.e. 0x029E (that means hexadecimal value 029E).

For signals on hardware products, an 'Active Low' is represented by the signal name with # following, i.e. IP\_RESET#.

Access terms are described as:

W Write Only

R Read Only

R/W Read/Write

R/C Read/Clear

R/S Read/Set

©2007 by TEWS TECHNOLOGIES GmbH

All trademarks mentioned are property of their respective owners.

---

<b>Issue</b>	<b>Description</b>	<b>Date</b>
1.0	First Issue	January 2007
1.1	Minor Changes in ACR	March 2007

# Table of Contents

<b>1</b>	<b>PRODUCT DESCRIPTION</b> .....	<b>8</b>
<b>2</b>	<b>TECHNICAL SPECIFICATION</b> .....	<b>10</b>
<b>3</b>	<b>FUNCTIONAL DESCRIPTION</b> .....	<b>11</b>
	<b>3.1 DMA Controller</b> .....	<b>11</b>
	3.1.1 DMAC Transmit Descriptor Lists.....	12
	3.1.1.1 Transmit Descriptor .....	12
	3.1.2 DMAC Receive Descriptor Lists.....	14
	3.1.2.1 Receive Descriptor .....	14
	3.1.2.2 Receive Data Section Status Byte (HDLC Mode) .....	16
	3.1.2.3 Receive Data Section Status Byte (ASYNC Modes).....	17
	<b>3.2 DMAC Interrupt Controller</b> .....	<b>18</b>
	3.2.1 Interrupt Vector Description .....	19
	3.2.1.1 Configuration Interrupt Vector .....	19
	3.2.1.2 DMA Controller Interrupt Vector .....	20
	3.2.1.3 SCC Interrupt Vector .....	22
<b>4</b>	<b>SERIAL COMMUNICATION CONTROLLER</b> .....	<b>23</b>
	<b>4.1 Protocol Description</b> .....	<b>23</b>
	4.1.1 HDLC Mode .....	24
	4.1.1.1 Address Mode 0 .....	24
	4.1.1.2 Extended Transparent Mode.....	24
	4.1.2 Asynchronous (ASYNC) Mode .....	24
	4.1.2.1 Asynchronous Mode.....	24
	4.1.2.2 Isochronous Mode.....	25
	4.1.2.3 Storage of Receive Data .....	25
	4.1.2.4 Data Transmission.....	25
	4.1.2.5 Break Detection .....	25
	4.1.2.6 Flow Control .....	26
	<b>4.2 Clock Sources</b> .....	<b>26</b>
	<b>4.3 Baud Rate Generation</b> .....	<b>28</b>
	<b>4.4 Data Encoding</b> .....	<b>28</b>
	4.4.1 NRZ and NRZI Encoding .....	28
	4.4.2 FM0 and FM1 Encoding.....	29
	4.4.3 Manchester Encoding .....	30
	<b>4.5 Clock Recovery (DPLL)</b> .....	<b>30</b>
<b>5</b>	<b>ADDRESS MAP</b> .....	<b>31</b>
	<b>5.1 PCI Configuration Space Registers</b> .....	<b>31</b>
	<b>5.2 PCI Base Address Register</b> .....	<b>31</b>
<b>6</b>	<b>REGISTER DESCRIPTION</b> .....	<b>32</b>
	<b>6.1 PCI Memory Space Registers Overview</b> .....	<b>32</b>
	<b>6.2 Global Registers Overview</b> .....	<b>33</b>
	<b>6.3 SCC Registers Overview</b> .....	<b>35</b>

<b>6.4</b>	<b>Global Registers.....</b>	<b>36</b>
6.4.1	GCMDBR – Global Command Register .....	36
6.4.2	GSTAR – Global Status Register.....	39
6.4.3	GMODE – Global Mode Register.....	40
6.4.4	IQLENR0 – Interrupt Queue Length Register 0.....	41
6.4.5	IQLENR1 – Interrupt Queue Length Register 1.....	41
6.4.6	IQSCCiRXBAR – Interrupt Queue SCCi Receiver Base Address Register .....	42
6.4.7	IQSCCiTXBAR – Interrupt Queue SCCi Transmitter Base Address Register.....	42
6.4.8	FIFOCCR4 – FIFO Control Register 4 .....	42
6.4.9	IQCFGBAR – Interrupt Queue Configuration Base Address Register .....	43
6.4.10	CHiCFG – Channel i Configuration Register .....	43
6.4.11	CHiBRDA – Channel i Base Receive Descriptor Address Register.....	45
6.4.12	CHiBTDA – Channel i Base Transmit Descriptor Address Register .....	46
6.4.13	CHiFRDA – Channel i First (Current) Receive Descriptor Address Register.....	46
6.4.14	CHiFTDA – Channel i First (Current) Transmit Descriptor Address Register .....	46
6.4.15	CHiLRDA – Channel i Last Receive Descriptor Address Register.....	46
6.4.16	CHiLTDA – Channel i Last Transmit Descriptor Address Register .....	47
6.4.17	VR – Version Register .....	47
6.4.18	ISPR – In-System-Programming Register .....	47
6.4.19	GCTLR – Global Control Register .....	47
<b>6.5</b>	<b>SCC Channel Specific Registers .....</b>	<b>48</b>
6.5.1	CMDR – Command Register .....	48
6.5.2	STAR – Status Register.....	48
6.5.3	CCR0 – Channel Configuration Register 0.....	49
6.5.4	CCR1 – Channel Configuration Register 1.....	50
6.5.5	CCR2 – Channel Configuration Register 2 .....	53
6.5.6	BRR – Baud Rate Register .....	55
6.5.7	TCR – Termination Character Register .....	55
6.5.8	IMR – Interrupt Mask Register.....	56
6.5.9	ISR – Interrupt Status Register.....	57
6.5.10	ACR – Additional Configuration Register .....	59
<b>7</b>	<b>CONFIGURATION HINTS .....</b>	<b>62</b>
7.1	Big / Little Endian.....	62
7.2	Configuration EEPROM.....	62
7.3	Additional Termination Resistors .....	64
<b>8</b>	<b>PIN ASSIGNMENT – I/O CONNECTOR .....</b>	<b>65</b>
8.1	Front Panel I/O Connector .....	65
8.2	Back I/O PMC Connector.....	67

## Table of Figures

FIGURE 1-1 : BLOCK DIAGRAM.....	9
FIGURE 2-1 : TECHNICAL SPECIFICATION.....	10
FIGURE 3-1 : FPGA ON-CHIP BLOCK DIAGRAM .....	11
FIGURE 3-2 : TRANSMIT DESCRIPTOR.....	12
FIGURE 3-3 : RECEIVE DESCRIPTOR .....	14
FIGURE 3-4 : CONFIGURATION INTERRUPT VECTOR.....	19
FIGURE 3-5 : DMA CONTROLLER INTERRUPT VECTOR .....	20
FIGURE 3-6 : SCC INTERRUPT VECTOR .....	22
FIGURE 4-1 : PROTOCOL MODES .....	23
FIGURE 4-2 : ADDRESS MODE 0 .....	24
FIGURE 4-3 : CLOCK SOURCES .....	27
FIGURE 4-4 : NRZ AND NRZI DATA ENCODING .....	29
FIGURE 4-5 : FM0 AND FM1 DATA ENCODING.....	29
FIGURE 4-6 : MANCHESTER DATA ENCODING .....	30
FIGURE 5-1 : PCI CONFIGURATION SPACE HEADER.....	31
FIGURE 5-2 : ADDRESS SPACE CONFIGURATION.....	31
FIGURE 6-1 : LOCAL REGISTER SPACE .....	32
FIGURE 6-2 : GLOBAL REGISTERS .....	34
FIGURE 6-3 : SCC REGISTERS .....	35
FIGURE 6-4 : GLOBAL COMMAND REGISTER.....	38
FIGURE 6-5 : GLOBAL STATUS REGISTER.....	40
FIGURE 6-6 : GLOBAL MODE REGISTER.....	40
FIGURE 6-7 : INTERRUPT QUEUE LENGTH REGISTER 0.....	41
FIGURE 6-8 : INTERRUPT QUEUE LENGTH REGISTER 1 .....	41
FIGURE 6-9 : IQSCCIRXBAR REGISTER .....	42
FIGURE 6-10: IQSCCITXBAR REGISTER.....	42
FIGURE 6-11 : FIFO CONTROL REGISTER 4 .....	43
FIGURE 6-12: IQCFGBAR REGISTER .....	43
FIGURE 6-13: CHICFG REGISTER .....	45
FIGURE 6-14: CHIBRDA REGISTER .....	45
FIGURE 6-15: CHIBTDA REGISTER .....	46
FIGURE 6-16: CHIFRDA REGISTER .....	46
FIGURE 6-17: CHIFTDA REGISTER.....	46
FIGURE 6-18: CHILRDA REGISTER .....	46
FIGURE 6-19: CHILTDA REGISTER.....	47
FIGURE 6-20: VERSION REGISTER.....	47
FIGURE 6-21: GLOBAL CONTROL REGISTER.....	47
FIGURE 6-22: COMMAND REGISTER .....	48
FIGURE 6-23: STATUS REGISTER .....	49
FIGURE 6-24: CHANNEL CONFIGURATION REGISTER 0.....	50

---

FIGURE 6-25: CHANNEL CONFIGURATION REGISTER 1.....	52
FIGURE 6-26: CHANNEL CONFIGURATION REGISTER 2.....	54
FIGURE 6-27: BAUD RATE REGISTER.....	55
FIGURE 6-28: TERMINATION CHARACTER REGISTER.....	55
FIGURE 6-29: INTERRUPT MASK REGISTER .....	56
FIGURE 6-30: INTERRUPT STATUS REGISTER .....	58
FIGURE 6-31: ADDITIONAL CONFIGURATION REGISTER .....	60
FIGURE 6-32: PHYSICAL INTERFACE MODE SELECTION .....	61
FIGURE 7-1 : CONFIGURATION EEPROM DATA .....	63
FIGURE 7-2: TERMINATION RESISTOR ASSIGNMENTS .....	64
FIGURE 7-3: TERMINATION RESISTOR POSITIONS.....	64
FIGURE 8-1 : FRONT PANEL I/O CONNECTOR NUMBERING .....	65
FIGURE 8-2 : FRONT I/O PIN ASSIGNMENT.....	66
FIGURE 8-3 : P14 I/O PIN ASSIGNMENT.....	67

# **1 Product Description**

The TPMC863 is a standard single-width 32 bit PMC with four high speed serial data communication channels. It is the successor of the discontinued TPMC862, providing similar functionality and full connector and pin-out compatibility.

The serial communication controller is implemented in FPGA logic, along with the bus master capable PCI interface, guaranteeing long term availability and having the option to implement additional functions in the future.

Each channel has receive and transmit FIFOs of 512 long words (32 bit) per channel for high data throughput.

Data transfer on the PCI bus is handled via TPMC863 initiated DMA cycles with minimum host/CPU intervention.

Several serial communication protocols are supported by each channel, such as asynchronous, isochronous, synchronous and HDLC mode.

A 14.7456 MHz oscillator provides standard asynchronous baud rates. A 24 MHz and a 10 MHz oscillator are provided for other (synchronous) baud rates.

Each channel provides various interrupt sources, generated on INTA. The interrupt sources can be enabled or disabled individually.

Multiprotocol transceivers are used for the line interface. The physical interface is selectable by software, individually for each channel as EIA-232, EIA-422, EIA 449, EIA-530, EIA-530A, V.35, V.36 or X.21.

The following signals are provided by the TPMC863 for each channel at the front panel I/O via an HD68 SCSI-3 type connector and at rear I/O via P14 connector:

Receive Data (RxD +/-), Transmit Data (TxD +/-), Receive Clock (RxC +/-), Transmit Clock (TxC +/-), Ready-To-Send (RTS +/-), Clear-To-Send (CTS +/-), Carrier-Detect (CD +/-) and GND. Additionally serial channel 3 provides Data-Set-Ready (DSR3 +/-) and Data-Terminal-Ready (DTR3 +/-) at the front I/O connector.

A serial EEPROM is used to store detailed board information by software and special configuration parameters.

For First-Time-Buyers the engineering documentation TPMC863-ED is recommended. The engineering documentation includes TPMC863-DOC, schematics and data sheets of TPMC863 devices.

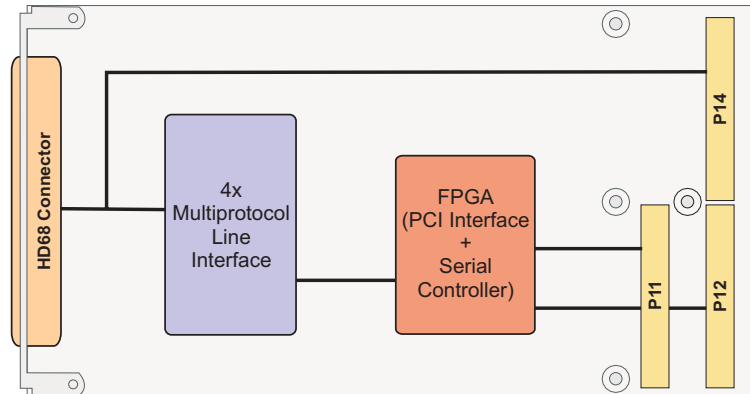


Figure 1-1 : Block Diagram

## 2 Technical Specification

<b>PMC Interface</b>	
<b>Mechanical Interface</b>	PCI Mezzanine Card (PMC) Interface Single Size
<b>Electrical Interface</b>	PCI Rev. 2.1 compliant 33 MHz / 32 bit PCI 3.3V and 5V PCI Signaling Voltage
<b>On Board Devices</b>	
<b>Serial Controller with Integrated PCI Interface</b>	XC3S1500-4FG(G)320I Spartan-3 FPGA (Xilinx) with PCI-Core (Xilinx)
<b>Serial Interface</b>	
<b>Number of Channels</b>	4
<b>Physical Interface</b>	LTC1544/LTC1546 (Linear Technology) Multiprotocol chip set, software-selectable, on-chip cable termination
<b>ESD Protection</b>	4kV
<b>FIFO</b>	Receive and transmit FIFOs with 512 DWORD per channel (16 KByte in total); additional 16 Byte SCC FIFOs per channel and direction
<b>Maximum Data Rate</b>	10 Mbit/s (synchronous) 2 Mbit/s (asynchronous) 2 Mbit/s (with clock recovery using DPLL)
<b>Interrupts</b>	Using PCI INTA for all channels, on board Interrupt Status Register
<b>I/O Connector</b>	Front panel HD68 SCSI-3 Type Connector (AMP 787082-7 or compatible) PMC P14 I/O (64 pin Mezzanine Connector)
<b>Physical Data</b>	
<b>Power Requirements</b>	300mA typical (no cable mode) @ +5V DC 900mA typical (V.35 mode) @ +5V DC
<b>Temperature Range</b>	Operating     -40°C to +85 °C Storage       -40°C to +125°C
<b>MTBF</b>	589.000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G <sub>B</sub> 20°C. The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation.
<b>Humidity</b>	5 – 95 % non-condensing
<b>Weight</b>	78 g

Figure 2-1 : Technical Specification

## 3 Functional Description

Data transfers for each direction are handled via PCI DMA transfer. The transfers are controlled via linked lists of descriptors. See the following chapter for a more detailed description of the descriptor and data structures.

The interrupt handling / operation concept is basically the same as on the TPMC862, e.g. the Interrupt Queue concept is maintained.

The FIFO structure is slightly different to the TPMC862. There is a common interrupt vector FIFO, however, there are dedicated data FIFOs for each channel and direction (not a common central FIFO that is splitted between the channels as on the TPMC862).

The register map and register structure in the PCI target space is basically the same as on the TPMC862.

The data is not swapped by the DMA Controller, it is always stored in little endian format.

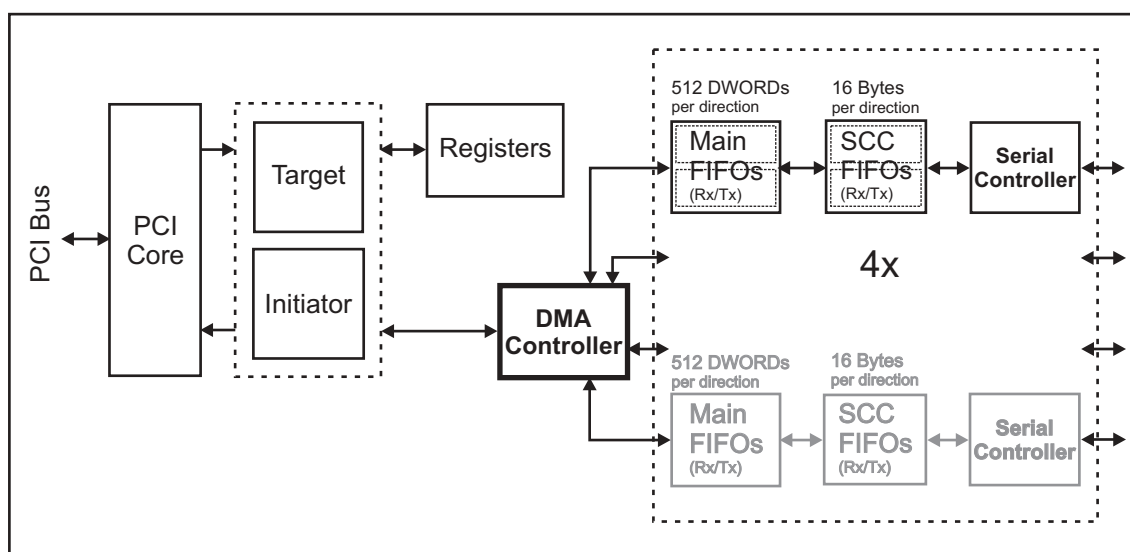


Figure 3-1 : FPGA On-Chip Block Diagram

### 3.1 DMA Controller

The CPU prepares linked lists for transmit and receive channels in the shared memory. These may be handled by dynamically allocating and linking descriptors and buffers as needed during runtime or by static predefined memory structures e.g. ring-chained-lists (the 'last' descriptor points back to the first descriptor). A mix of predefined descriptor lists but dynamically handled data buffers may also be an appropriate solution. This strategy depends on the specific application. The DMAC provides multiple control mechanisms supporting all of these combinations in an efficient way.

The descriptors and data buffers can be stored in separate memory spaces within the 32-bit address range allowing full scatter/gather methods of assembling and disassembling of packets.

Each descriptor contains a 'next descriptor address' field to implement the linked list. Because the DMA controller cannot distinguish between valid and invalid addresses, a 'Hold' mechanism is needed to prevent the DMA controller from branching to invalid memory locations.

Two alternative control mechanisms are provided to detect and handle descriptor list end (Hold) conditions:

- Hold bit control mode
- Last descriptor address control mode

The Control Mode applies to all DMA channels transmit and receive and is selected via bit 'CMODE' in Global Mode Register GMODE.

An HDLC frame may fit in one buffer connected to one descriptor or it may be split to several buffers each associated with linked descriptors. A 'frame end' indication (FE bit) will be set in each descriptor which points to the last buffer of one HDLC frame.

The 'frame end' indications are stored in the internal FIFOs influencing the FIFO control mechanisms. Therefore 'frame end' indications (FE bit) are also used in non frame oriented protocol modes such as ASYNC mode. They are referred to as 'frame end/block end' indication in the following chapters.

### 3.1.1 DMAC Transmit Descriptor Lists

Each transmit descriptor consists of 4 consecutive DWORDs located DWORD aligned in the shared memory. The first 3 DWORDs are written by the host and read by the corresponding DMA channel using a burst transaction, when requested to do by the host either via an 'AR' (Action Request) command or a transmit poll command or after branching from previous transmit descriptors. They provide information about the next descriptor in the linked list, the attached transmit data buffer and its size, as well as some control bits.

The fourth DWORD is written by the DMA channel indicating that operation on this descriptor is finished.

The CPU will write the address of the first descriptor of each linked list to a dedicated Base Address Register (BTDAi) during initialization procedure. The corresponding DMA channel starts serving the descriptor at these addresses.

#### 3.1.1.1 Transmit Descriptor

DWORD	31	30	29	28..16	15..0
0	FE	HOLD	HI	NO	0
1	Next Transmit Descriptor Pointer				
2	Transmit Data Pointer				
3	0	C	0		

Figure 3-2 : Transmit Descriptor

#### FE: Frame End, set by the host

It indicates that the current transmit data section (addressed by Transmit Data Pointer) contains the end of a frame (HDLC) or the end of data block (ASYNC). When transferring the last data from this transmit data section into the internal FIFO the DMAC marks this data with a 'frame end / block end' indication bit.

**GMODE.CMOD='0':**

After that it checks the HOLD bit. If HOLD='0', it branches to the next transmit descriptor. Otherwise the corresponding DMAC transmit channel is deactivated as long as the host CPU does not request reactivation via the GCMR register (either transmit poll request or action request with 'IDT' command).

**GMODE.CMOD='1':**

After that it checks if the first (current) transmit descriptor address (LTDA) is equal to the last transmit descriptor address (LTDA) stored in the corresponding channel specific on-chip registers. When both addresses differ, it branches to the next transmit descriptor. Otherwise the corresponding DMAC transmit channel is deactivated as long as the host CPU does not write a new LTDA value to LTDA register or provides an action request with 'IDT' command.

**HOLD: Hold (only valid when GMODE.CMODE=0)**

It indicates whether the current descriptor is the last element of a linked list or not:

**HOLD='0':**

A next descriptor is available in the shared memory; after checking the HOLD bit stored in the on-chip memory the DMAC branches to next transmit descriptor.

**HOLD='1':**

The current descriptor is the last one that is available for the DMAC. The corresponding DMAC channel is deactivated for transmit direction as long as the microprocessor does not request an activation via the CMDR register.

**NO: Byte Number**

This byte number defines the number of bytes stored in the data section to be transmitted. Thus the maximum length of data buffer is 8191 bytes (i.e. NO = 0x1FFF). A transmit descriptor and the corresponding data section must contain at least either one data byte or a frame end indication. Otherwise a DMA controller interrupt with 'ERR' bit set is generated.

**HI: Host Initiated Interrupt**

If the HI bit is set, the corresponding DMAC generates an interrupt with set HI bit after transferring all data bytes of the current data section.

**Next Transmit Descriptor Pointer:**

This 32-bit pointer contains the start address of the next transmit descriptor. After sending the indicated number of data bytes, the DMAC branches to the next transmit descriptor to continue transmission. The transmit descriptor is read entirely at the beginning of transmission and stored in on-chip memory. Therefore all information in the next descriptor must be valid when the DMAC branches to this descriptor. This pointer is not used if a transmitter reset or initialization channel command is detected while the DMAC still reads data from the current transmit descriptor. In this case BTDA value in the BTDA register is used as a pointer for the next transmit descriptor to be branched to.

**Transmit Data Pointer:**

This 32-bit pointer contains the start address of the transmit data section. Although the TPMC863 works long word oriented, it is possible to begin transmit data section at byte addresses.

**C: Complete**

This bit is set by the DMAC if

- it completes reading data section normally
- it was aborted by a transmitter reset command.

**3.1.2 DMAC Receive Descriptor Lists**

Each receive descriptor consists of 5 consecutive DWORDs located DWORD aligned in the shared memory. The first 3 DWORDs are read by the corresponding DMA channel using a burst transaction and provide information about the next descriptor in the linked list, the attached receive data buffer and its size, as well as some control bits.

The fourth DWORD is written by the DMA channel indicating that operation on this descriptor is finished. The fifth DWORD is also written by the DMA channel but only in descriptors containing the first or only data section of an HDLC frame or data block. It is a pointer to the last descriptor containing the frame or block end ('FE' bit) allowing the software to unchain the complete partial descriptor list containing one frame or block without parsing through the list for 'FE' indication.

The CPU will write the address of the first descriptor of each linked list to a dedicated Base Address Register during initialization procedure. The corresponding DMA channels start operating the linked lists at these addresses.

**3.1.2.1 Receive Descriptor**

DWORD	31	30	29	28..16	15..8	7..0
0	0	HOLD	HI	NO	0	
1	Next Receive Descriptor Pointer					
2	Receive Data Pointer					
3	FE	C	0	BNO	STATUS	0
4	Frame End Descriptor Pointer					

Figure 3-3 : Receive Descriptor

**HOLD: Hold** (only valid when GMODE.CMODE=0)

It indicates whether the current descriptor is the last element of a linked list or not:

HOLD='0':

A next descriptor is available in the shared memory; after checking the HOLD bit stored in the on-chip memory the DMAC branches to next receive descriptor

HOLD='1':

---

The current descriptor is the last one that is available for the DMAC. After completion of the current receive descriptor an interrupt is generated and the corresponding DMAC channel is deactivated for receive direction as long as the microprocessor does not request an activation via the CMDR register.

**HI: Host Initiated Interrupt**

If the HI bit is set, the corresponding DMAC generates an interrupt with set HI bit after transferring all data bytes into the current data section.

**NO: Byte Number**

This byte number defines the size of the receive data section allocated by the host. It has to be a multiple of 4 bytes which is responsibility of the software. The maximum buffer length is 8188 bytes (i.e. NO = 0x1FFC).

**Next Receive Descriptor Pointer:**

This 32-bit pointer contains the start address of the next receive descriptor. After completing the current receive descriptor the DMAC branches to the next receive descriptor to continue reception. The receive descriptor is read entirely at the beginning of reception and stored in on-chip memory. Therefore all information in the next descriptor must be valid when the DMAC branches to this descriptor.

**Receive Data Pointer:**

This 32-bit 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 Receive Data Pointer) contains the end of a frame (HDLC) or the end of data block (ASYNC). This bit is set by the DMAC after transferring the last data from the internal FIFO into the receive data section. Moreover the BNO and STATUS is updated and the 'C' bit is set by the DMAC.

GMODE.CMODE='0':

After that it checks the HOLD bit stored in the on-chip memory. If HOLD='0', it branches to the next receive descriptor. Otherwise the corresponding DMAC receive channel is deactivated as long as the host CPU does not request reactivation via the GCMDR register (action request with 'IDR' command).

GMODE.CMODE='1':

After that it checks if the first (current) receive descriptor address (LRDA) is equal to the last receive descriptor address (LRDA) stored in the corresponding channel specific on-chip registers. When both addresses differ, it branches to the next receive descriptor. Otherwise the corresponding DMAC receive channel is deactivated as long as the host CPU does not write a new LRDA value to LRDA register or provides an action request with 'IDR' command.

**C: Complete**

This bit is set by the DMAC if:

- it completed filling data section normally
- it was aborted by a receiver reset command
- end of frame (HDLC) or end of block (ASYNC) was stored in the receive data section.

**BNO: Byte Number of Received Data**

DMAC writes the number of data bytes that were stored in the current data section into BNO.

**Frame End Descriptor Pointer:**

This 32-bit pointer is only valid in the descriptor, which contains the data pointer to the first data section of an HDLC frame or ASYNC block. This pointer is updated by the DMAC with the address of the descriptor that contains the data pointer to the last section (FE) of the HDLC frame or ASYNC block.

**Receive descriptor STATUS bit field:**

7	6	5	4	3	2	1	0
0	0	0	0	0	0	RA	0

**RA: Receive Abort**

This bit indicates that the reception of a frame (HDLC) or block (ASYNC) was ended by a DMA receiver reset command or by a HOLD bit in the current receive descriptor or by a FRDA=LRDA condition.

**3.1.2.2 Receive Data Section Status Byte (HDLC Mode)**

In HDLC protocol mode, the last byte of a frame (Receive Status Byte, RSTA) - located in the data section - contains error indications caused by the SCC (e.g. CRC, receive abort, ...).

**RSTA:**

7	6	5	4	3	2	1	0
1	RDO	CRC	RAB	0	0	0	0

The contents of the RSTA byte relates to the received HDLC frame and is generated when end-of-frame is recognized at the serial receive interface. Bit 7 is always '1' for compatibility with TPMC862.

**RDO: Receive Data Overflow**

A data overflow has occurred during reception of the frame. Additionally, an interrupt can be generated (refer to ISR.RDO / IMR.RDO).

**CRC: CRC Compare/Check**

0: CRC check failed, received frame contains errors.

1: CRC check OK, no errors detected in received frame.

**RAB: Receive Message Aborted**

The received frame was aborted from the transmitting station. According to the HDLC protocol, this frame must be discarded by the receiver station.

**3.1.2.3 Receive Data Section Status Byte (ASYNC Modes)**

In ASYNC protocol mode a status byte can be attached additionally to every stored data byte (CCR2.RFDF='1').

The data character and status character format is determined as follows:

<b>15</b>	<b>14</b>	<b>13..9</b>	<b>8</b>	<b>7..0</b>
parity error	frame error	reserved	parity bit	data byte

## 3.2 DMAC Interrupt Controller

The interrupt concept is based on 32-bit interrupt vectors generated by the different blocks. Interrupt vectors are stored in an interrupt FIFO which is 32 DWORDs deep. The interrupt controller transfers available vectors to one of nine circular interrupt queues located in the shared memory depending on the source ID of each interrupt vector.

In addition new interrupt vectors are indicated in the global status register GSTAR on a per queue basis and selectively confirmed by writing '1' to the corresponding GSTAR bit positions. The PCI interrupt signal INTA is asserted with any new interrupt event and remains asserted until all events are confirmed.

Each interrupt queue length and memory location can be configured via specific interrupt queue base address registers and two shared interrupt queue length registers. The queue length is individually programmable in multiples of 32 DWORDs (see IQLENR0/1).

One dedicated interrupt queue is provided per SCC channel and direction (IQSCCiRX and IQSCCiTX). Non channel specific interrupt vectors generated by the DMAC itself are transferred to the configuration queue IQCFG.

The internal blocks provide mask registers for suppressing interrupt indications. Masked interrupts will neither generate an interrupt vector nor an INTA signal and GSTAR indication.

The DMA interrupt controller generates 6 channel specific interrupts regarding transmit and receive descriptor handling:

### Host Initiated interrupt (HI):

This interrupt can be forced by setting bit 'HI' in the receive and transmit descriptor. In this case the DMAC will generate an HI-interrupt with completion of this descriptor i.e. when the DMAC is ready to branch to the next descriptor address. This might be used to monitor the progress of the corresponding DMA channel on the descriptor list. As an example the HI interrupt can be used to dynamically request attachment of new receive descriptors to the list if the DMA channel comes close to the list end.

### Frame Indication interrupt (FI):

This interrupt is generated with completion of any receive and transmit descriptor with a set 'frame end/block end' indication, i.e. FE='1'.

### Error interrupt (ERR):

Indicates an unexpected descriptor configuration

Receive descriptor:

ERR is generated if receive data cannot be transferred to the shared memory completely because the frame (block) does not fit into the current data section and a HOLD condition (HOLD bit or LRDA=FRDA) prevents the DMAC from branching to the next descriptor.

ERR is also generated if an already started DMA transfer is aborted by a receive DMA reset (RDR) command.

Transmit descriptor:

In transmit direction an ERR interrupt is generated if one of the following descriptor settings is detected

- HOLD='1' and FE='0' (the already started transmit frame could not be finished)
- LTDA=FTDA and FE='0' (the already started transmit frame could not be finished)
- FE='0' and NO='0' (a packet of length 0 is supposed to be a 'frame' with FE bit set)

The DMA controller will continue 'normal' operation in case of an ERR event. Nevertheless these cases may result in receive data overflows or transmit data underruns.

FI and HI interrupt indications caused by one descriptor will be generated into one interrupt vector with 'HI' and 'FI' bit set.

### 3.2.1 Interrupt Vector Description

#### 3.2.1.1 Configuration Interrupt Vector

Configuration interrupt vectors are transferred to Configuration Interrupt Queue 'IQCFG'.

31..28	27..2	1	0
Source ID = 1010	0	ARF	ARACK

Figure 3-4 : Configuration Interrupt Vector

#### **ARF: Action Request Failed Interrupt**

This bit indicates that an action request command was completed with an 'action request failed' condition:

ARF='0': No action request was performed or no 'action request failed' condition occurred completing an action request.

ARF='1': The last action request command was completed with an 'action request failed' condition.

#### **ARACK: Action Request Acknowledge Interrupt**

This bit indicates that an action request command was completed successfully:

ARACK='0': No action request was performed or completed successfully.

ARACK='1': The last action request command was completed successfully.

### 3.2.1.2 DMA Controller Interrupt Vector

DMA controller interrupt vectors are transferred to the corresponding channel and direction specific interrupt queues IQSCCiRX and IQSCCiTX respectively.

<b>31</b>	<b>30..28</b>	<b>27..19</b>	<b>18</b>	<b>17</b>	<b>16</b>	<b>15..0</b>
0	Source ID	0	HI	FI	ERR	0

Figure 3-5 : DMA Controller Interrupt Vector

**Source ID:** 000 Receive Channel 0 Interrupt Vector (IQSCC0RX)

001 Receive Channel 1 Interrupt Vector (IQSCC1RX)

010 Receive Channel 2 Interrupt Vector (IQSCC2RX)

011 Receive Channel 3 Interrupt Vector (IQSCC3RX)

100 Transmit Channel 0 Interrupt Vector (IQSCC0TX)

101 Transmit Channel 1 Interrupt Vector (IQSCC1TX)

110 Transmit Channel 2 Interrupt Vector (IQSCC2TX)

111 Transmit Channel 3 Interrupt Vector (IQSCC3TX)

#### HI: Host Initiated interrupt (Rx/Tx Channel)

This bit indicates that a Host Initiated (HI) interrupt occurred, i.e. the corresponding DMA controller channel detects the 'HI' bit set to '1' in the receive or transmit descriptor before branching to the next descriptor.

HI='0' No Host Initiated (HI) interrupt is indicated by this vector.

HI='1' A Host Initiated (HI) interrupt is indicated by this vector.

#### FI: Frame Indication interrupt (Rx/Tx Channel)

This bit indicates that a Frame Indication (FI) interrupt occurred.

##### Receive direction:

FI='1' indicates, that a frame has been received completely or was stopped by a DMAC receiver reset command or a hold condition set in a receive descriptor. It is set when the DMAC branches from the last descriptor belonging to the current frame (or block) (FE='1') to the first descriptor of a new frame. It is also set when the descriptor in which the frame/block is finished contains a hold condition.

Transmit direction:

Issued if the 'FE' bit is detected in the transmit descriptor. It is set when the DMAC branches to the next transmit descriptor, belonging to a new frame or when 'HOLD' bit is set in conjunction with 'FE' bit. Only 'ERR' indication without 'FI' is set, if a transmit descriptor contains a 'HOLD' (hold condition) but no 'FE' bit.

FI='0' No Frame Indication (FI) interrupt is indicated by this vector.

FI='1' A Frame Indication (FI) interrupt is indicated by this vector.

**ERR: ERROR Indication interrupt (Rx/Tx Channel)**

This bit indicates that an Error interrupt occurred.

Receive direction:

Issued if the current frame/block could not be transferred to the shared memory completely, because of a hold condition in a receive descriptor not providing enough bytes for the frame/block or the frame/block was aborted by a DMAC receiver reset command.

Transmit direction:

Issued if a transmit descriptor contains a hold condition but FE='0' or if the last descriptor had NO=0 and FE='0'.

ERR='0' No Error (ERR) interrupt is indicated by this vector.

ERR='1' An Error (ERR) interrupt is indicated by this vector.

### 3.2.1.3 SCC Interrupt Vector

Serial Channel (SCC) related interrupt vectors are transferred to the corresponding channel and direction specific interrupt queues IQSCCiRX and IQSCCiTX respectively.

**Interrupt vectors generated by the SCCs might contain interrupt indications for both, receive AND transmit direction. But in receive interrupt queues only the receive interrupt indications need to be served and in transmit interrupt queues only transmit interrupt indications need to be served by the software.**

31	30..28	27..24	23..19	18	17	16
0	Source ID	0010	0	ALLS	0	XDU

15	14	13..10	9	8	7	6	5	4	3	2	1	0
0	CSC	0	BRK	BRKT	TCD	TIME	PERR	FERR	PLLA	CDSC	RFO	0

Figure 3-6 : SCC Interrupt Vector

**Source ID:** 000 Receive Channel 0 Interrupt Vector (IQSCC0RX)

001 Receive Channel 1 Interrupt Vector (IQSCC1RX)

010 Receive Channel 2 Interrupt Vector (IQSCC2RX)

011 Receive Channel 3 Interrupt Vector (IQSCC3RX)

100 Transmit Channel 0 Interrupt Vector (IQSCC0TX)

101 Transmit Channel 1 Interrupt Vector (IQSCC1TX)

110 Transmit Channel 2 Interrupt Vector (IQSCC2TX)

111 Transmit Channel 3 Interrupt Vector (IQSCC3TX)

Bit field 18..0 of the SCC interrupt vector is a copy of the SCC Interrupt Status Register ISR (for detailed information see chapter 'ISR - Interrupt Status Register').

# 4 Serial Communication Controller

## 4.1 Protocol Description

The following table provides an overview of all supported protocol modes and their assignment to the major protocol engines HDLC and ASYNC. The protocol engine of each SCC is selected via bit field 'SM' in register CCR0. The HDLC Sub Modes are selected via additional bit fields in registers CCR0 and CCR1.

<b>Protocol Engine:</b>	<b>Protocol Mode:</b>	
<b>Register CCR0 Setting:</b>		<b>Register CCR1 Setting (bit fields MDS):</b>
HDLC SM = '00'	HDLC address mode 0	MDS = '10'
	Extended transparent mode	MDS = '11'
		<b>Register CCR0 Setting (bit BCR):</b>
ASYNC SM = '11'	Asynchronous mode	BCR = '1'
	Isochronous mode	BCR = '0'

Figure 4-1 : Protocol Modes

**Extended transparent is a fully bit-transparent transmit/reception mode which is treated as a sub-mode of the HDLC block.**

## 4.1.1 HDLC Mode

### 4.1.1.1 Address Mode 0

Standard HDLC framing is performed and each complete frame will be stored in the receive FIFO, optional CRC generation/check.

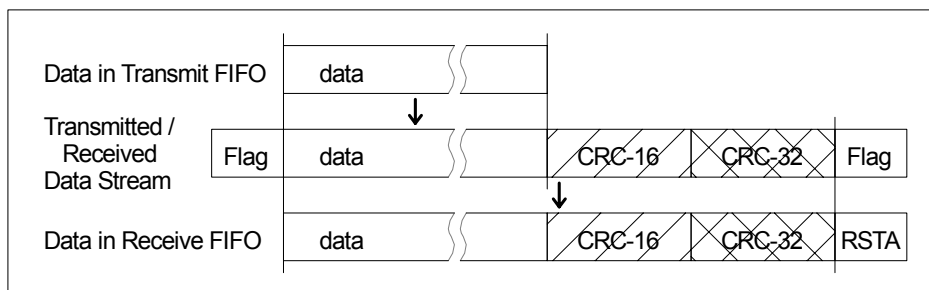


Figure 4-2 : Address Mode 0

### 4.1.1.2 Extended Transparent Mode

In extended 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 allows user specific protocol variations.

## 4.1.2 Asynchronous (ASYNC) Mode

Character framing is achieved by start and stop bits. Each data character is preceded by one start bit and terminated by one or two stop bits. The character length is selectable from 5 up to 8 bits. Optionally, a parity bit can be added which complements the number of ones to an even or odd quantity (even/odd parity). The parity bit can also be programmed to have a fixed value (Mark or Space). The character format configuration is performed via appropriate bit fields in register CCR2.

### 4.1.2.1 Asynchronous Mode

Bit clock rate 16 is selected (register CCR0, bit BCR = '1'), NRZ data encoding should be selected (register CCR0, bit field 'SC').

The receiver operates at a clock rate which is 16 times the nominal (expected) data bit rate. It synchronizes itself to each character by detecting and verifying the start bit. Oversampling (3 samples) around the nominal bit center in conjunction with majority decision is provided for every received bit (including start bit).

The synchronization lasts for one character; the next incoming character causes a new synchronization. As a result, the demand for high clock accuracy is reduced. Two communication stations using the asynchronous procedure are clocked independently; their clocks need not to be in phase or locked to exactly the same frequency but, in fact, may differ from one another within a certain range.

#### 4.1.2.2 Isochronous Mode

Bit clock rate 1 is selected (register CCR0 bit BCR = '0').

The isochronous mode uses the asynchronous character format. However, each data bit is only sampled once (no oversampling). The input clock has to be externally phase locked to the data stream. This mode allows much higher transfer rates.

#### 4.1.2.3 Storage of Receive Data

If the receiver is enabled, received data is stored in the receive FIFO (the LSB is received first). Character length, number of stop bits and the optional parity bit are checked. Errors are indicated via interrupts. Additionally, the character specific error status (framing and parity) can optionally be stored in the receive FIFO. Filling of the receive FIFO is controlled by

- a programmable threshold level (bit field 'RFTH' in register CCR2),
- the selected data format (bit 'RFDF' in register CCR2),
- detection of the programmable Termination Character (bit 'TCDE' and bit field 'TC' in register TCR).

Additionally, the time-out event interrupt as an optional status information indicates that a certain time (refer to register CCR1) has elapsed since the reception of the last character.

#### 4.1.2.4 Data Transmission

The selection of asynchronous or isochronous operation has no further influence on the transmitter. The bit clock rate is solely a dividing factor for the selected clock source.

Transmission of the contents of the transmit FIFO starts after providing data to the DMA controller. The character frame for each character, consisting of start bit, the character itself with defined character length, optionally generated parity bit and stop bit(s) is assembled.

After finishing transmission (indicated by the 'ALLS' interrupt), IDLE sequence (logical '1') is transmitted on transmit pin TxD.

Additionally, the CTS signal may be used to control data transmission.

#### 4.1.2.5 Break Detection

The SCC recognizes the break condition upon receiving consecutive (physical) '0's for the defined character length, the optional parity and the selected number of stop bits ('zero' character and framing error). The 'zero' character is not pushed to RFIFO. The 'Break' interrupt (BRK) is generated, if enabled. The break condition will be present until a '1' is received which is indicated by the 'Break Terminated' interrupt (BRKT).

### 4.1.2.6 Flow Control

#### Transmitter:

The transmitter output is enabled if CTS signal is 'LOW'. Setting bit CCR1.FCTS = '1' allows the transmitter to send data independent of the condition of the CTS signal.

#### Receiver:

For some applications it is desirable to provide means of flow control to indicate to the far end transmitter that the local receiver's buffer is getting full.

This flow control can be used between two DTEs and between a DTE and a DCE (MODEM) that supports this kind of bi-directional flow control.

Setting bit CCR1.FRTS = '1' and CCR1.RTS = '0' invokes the flow control for the receiver. When the SCC receive FIFO has reached a threshold of 12 bytes (16 bytes total), the RTS signal is forced inactive (HIGH). When the receive FIFO is nearly empty (2 bytes), the RTS is re-asserted ('LOW'). Note that data is immediately transferred from the SCC receive FIFO to the DMA accessible FIFO (as long as there is space available). Thus when the SCC receive FIFO reaches the 12 bytes threshold, there are 4 more bytes storage available before an overflow can occur. This provides sufficient time for the far end transmitter to react to the change in the RTS signal and stop sending more data.

## 4.2 Clock Sources

The TPMC863 has several clock sources for transmitter and receiver controlled by the ACR. Possible sources are the three on-board oscillators (14.7456 MHz, 24 MHz and 10 MHz), the external RxCLK and TxCLK inputs. TxCLK can be an input for the internal transmit clock or an output for transmit clock monitor signal (see following figure).

The guaranteed maximum data rate is limited by the multiprotocol transceivers and is at least 10Mbit/s in EIA-530 (V.11) and V.35 transceiver modes. Maximum data rate of EIA-232 (V.28) is 115.2kbit/s. (see datasheets of transceivers, e.g. in the TPMC863-ED). The maximum data rate in asynchronous mode with oversampling or DPLL mode is 2 Mbit/s. To generate higher internal clock frequencies for oversampling or DPLL reference clock, an optional x4 clock multiplier can be used.

**The input frequency range of the x4 clock multiplier is 4.5 MHz to 28 MHz, these values must never be exceeded to ensure proper function of the clock multiplier.**

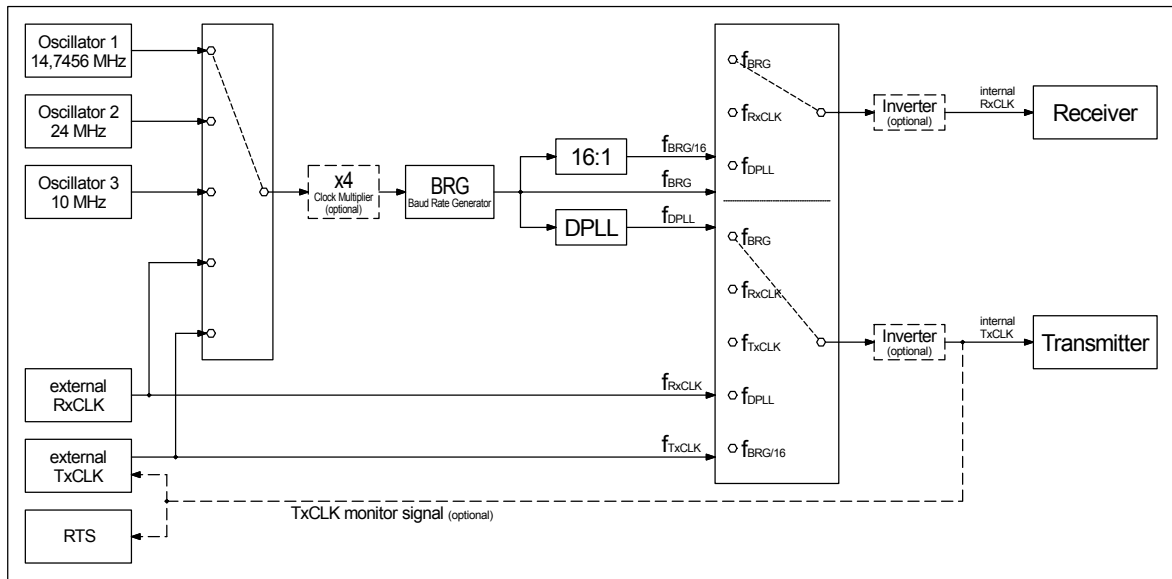


Figure 4-3 : Clock Sources

## 4.3 Baud Rate Generation

Each of the four channels has its own Baud Rate Generator (BRG). They are controlled by registers BRR (0x012C, 0x01AC, 0x022C, 0x02AC).

The Baud Rate Frequency is:  $f_{BRG} = f_{in} / k$

with the divisor:  $k = (N + 1) \times 2^M$

N (BRR[5:0]) = 0..63

M (BRR[11:8]) = 0..15

**When asynchronous mode is selected (register CCR0, bit BCR = '1'), k is multiplied by 16.**

## 4.4 Data Encoding

The following codings of the serial data are supported:

- Non-Return-To-Zero (NRZ)
- Non-Return-To-Zero-Inverted (NRZI)
- FM0 (known as Bi-Phase Space)
- FM1 (known as Bi-Phase Mark)
- Manchester (known as Bi-Phase)

### 4.4.1 NRZ and NRZI Encoding

**NRZ:** The signal level corresponds to the value of the data bit.

**NRZI:** A logical '0' is indicated by a transition and a logical '1' by no transition at the beginning of the bit cell.

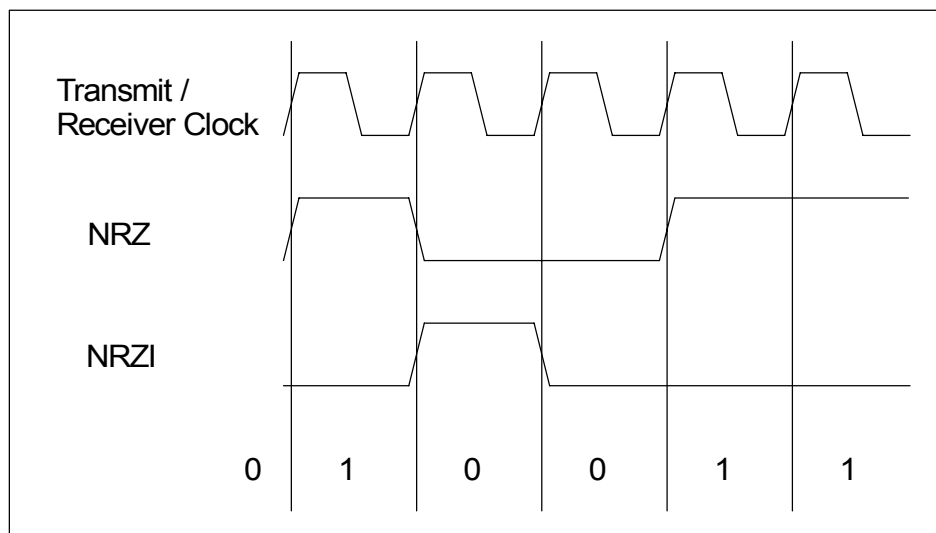


Figure 4-4 : NRZ and NRZI Data Encoding

#### 4.4.2 FM0 and FM1 Encoding

**FM0:** An edge occurs at the beginning of every bit cell. A logical '0' has an additional edge in the center of the bit cell, whereas a logical '1' has none. The transmit clock precedes the receive clock by 90°.

**FM1:** An edge occurs at the beginning of every bit cell. A logical '1' has an additional edge in the center of the bit cell, a logical '0' has none. The transmit clock precedes the receive clock by 90°.

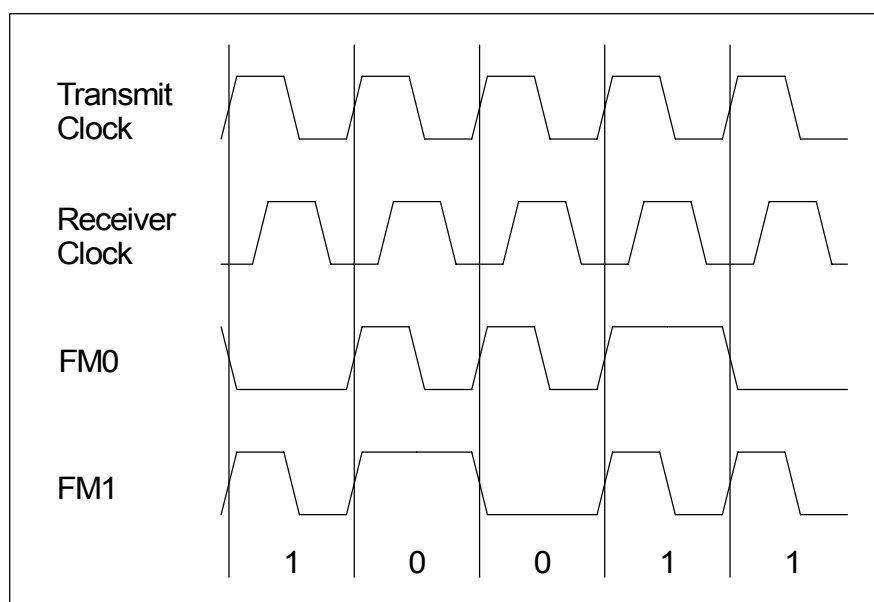


Figure 4-5 : FM0 and FM1 Data Encoding

### 4.4.3 Manchester Encoding

In the first half of the bit cell, the physical signal level corresponds to the logical value of the data bit. At the center of the bit cell this level is inverted. The transmit clock precedes the receive clock by 90°.

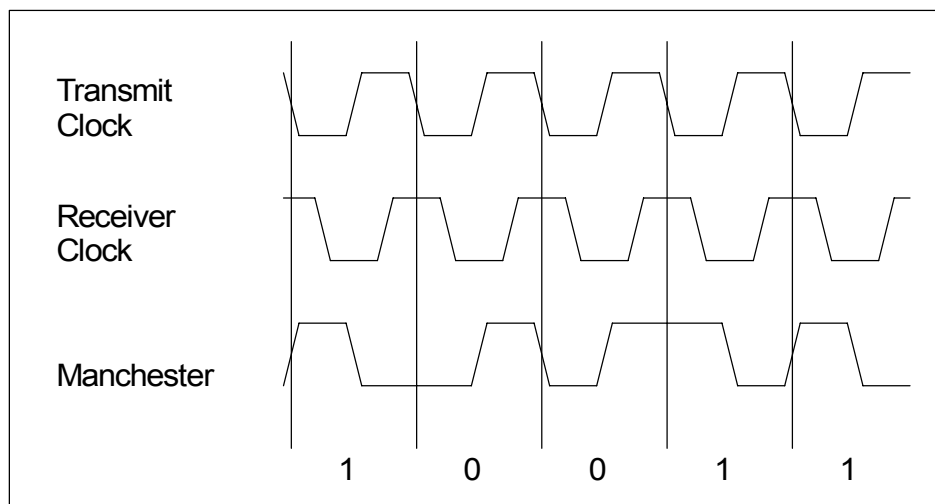


Figure 4-6 : Manchester Data Encoding

## 4.5 Clock Recovery (DPLL)

The SCC offers the advantage of recovering the received clock from the received data by means of internal DPLL circuitry, eliminating the need to transfer additional clock information via a separate serial clock line. For this purpose, the DPLL is supplied with a 'reference clock' from the BRG which is 16 times the expected data clock rate. The transmit clock may be obtained by dividing the output of the BRG by a constant factor of 16 or also directly from the DPLL (see 'TCS' in register ACR).


The main task of the DPLL is to derive a receive clock and to adjust its phase to the incoming data stream in order to enable optimal bit sampling.

The mechanism for clock recovery depends on the selected data encoding (see chapter "Data Encoding").

# 5 Address Map

## 5.1 PCI Configuration Space Registers

PCI Configuration Space Header				
Offset	31	16	15	0
0x00	Device ID (TPMC863: 0x035F)		Vendor ID (TEWS Technologies 0x1498)	
0x04	Status		Command	
0x08	Class Code (0x028000)			Rev ID
0x0C	BIST	Header Type (0x00)	Lat. Timer	Line Size
0x10	Base Address Register 0 (Target register space of the high speed serial controller)			
0x14	Base Address Register 1			
0x18	Base Address Register 2			
0x1C	Base Address Register 3			
0x20	Base Address Register 4			
0x24	Base Address Register 5			
0x28	CardBus CIS Pointer			
0x2C	Subsystem ID (s.b.)		Subsystem Vendor ID (TEWS Technologies 0x1498)	
0x30	Expansion ROM Base Address			
0x34	Reserved			Cap. Pointer (0x00)
0x38	Reserved			
0x3C	Max_Lat (0x0A)	Min_Gnt (0x03)	Int_Pin (0x01)	Int_Line

 = Not supported

Subsystem ID:	TPMC863-10 0x000A
---------------	-------------------

Figure 5-1 : PCI Configuration Space Header

## 5.2 PCI Base Address Register

Base Address (Offset in PCI Configuration Space)	PCI Space Mapping	Size (Byte)	Port Width (Bit)	Endian Mode	Description
0 (0x10)	MEM	2048	32	Little	Register Address Space

Figure 5-2 : Address Space Configuration

# 6 Register Description

## 6.1 PCI Memory Space Registers Overview

Offset to Base Address BAR0	Addresses range	Number of used DWORD registers	Description
0x0000	0x0000...0x00FF	44 (0x0000...0x00EC)	Global Registers
0x0100	0x0100...0x017F	10 (0x0100...0x0158)	SCC0 Registers
0x0180	0x0180...0x01FF	10 (0x0180...0x01D8)	SCC1 Registers
0x0200	0x0200...0x027F	10 (0x0200...0x0258)	SCC2 Registers
0x0280	0x0280...0x02FF	10 (0x0280...0x02D8)	SCC3 Registers
0x0300	0x0300...0x07FF	0	(reserved)

Figure 6-1 : Local Register Space

## 6.2 Global Registers Overview

All registers are 32 bit organized.

Offset to Base Address 0x0000 of Local Space	Register Name	
0x0000	GCMDR	Global Command Register
0x0004	GSTAR	Global Status Register
0x0008	GMODE	Global Mode Register
Interrupt Queue IQ specific registers:		
0x000C	IQLENR0	IQ Length Register 0
0x0010	IQLENR1	IQ Length Register 1
0x0014	IQSCC0RXBAR	IQ SCC0 RX Base Address Register
0x0018	IQSCC1RXBAR	IQ SCC1 RX Base Address Register
0x001C	IQSCC2RXBAR	IQ SCC2 RX Base Address Register
0x0020	IQSCC3RXBAR	IQ SCC3 RX Base Address Register
0x0024	IQSCC0TXBAR	IQ SCC0 TX Base Address Register
0x0028	IQSCC1TXBAR	IQ SCC1 TX Base Address Register
0x002C	IQSCC2TXBAR	IQ SCC2 TX Base Address Register
0x0030	IQSCC3TXBAR	IQ SCC3 TX Base Address Register
0x0034	FIFO CR4	FIFO Control Register 4
0x0038	reserved	-
0x003C	IQCFG BAR	IQ CFG Base Address Register
0x0040	reserved	-
0x0044	reserved	-
0x0048	reserved	-
0x004C	reserved	-
DMA Controller (DMAC) specific registers:		
0x0050	CH0CFG	Channel 0 Configuration Register
0x0054	CH0BRDA	Channel 0 Base Rx Descr. Address
0x0058	CH0BTDA	Channel 0 Base Tx Descr. Address
0x005C	CH1CFG	Channel 1 Configuration Register
0x0060	CH1BRDA	Channel 1 Base Rx Descr. Address
0x0064	CH1BTDA	Channel 1 Base Tx Descr. Address
0x0068	CH2CFG	Channel 2 Configuration Register
0x006C	CH2BRDA	Channel 2 Base Rx Descr. Address
0x0070	CH2BTDA	Channel 2 Base Tx Descr. Address
0x0074	CH3CFG	Channel 3 Configuration Register
0x0078	CH3BRDA	Channel 3 Base Rx Descr. Address

Offset to Base Address 0x0000 of Local Space	Register Name	
	0x007C	CH3BTDA
0x0080...0x0097	reserved	-
0x0098	CH0FRDA	Channel 0 First Rx Descr. Address
0x009C	CH1FRDA	Channel 1 First Rx Descr. Address
0x00A0	CH2FRDA	Channel 2 First Rx Descr. Address
0x00A4	CH3FRDA	Channel 3 First Rx Descr. Address
0x00A8	reserved	-
0x00AC	reserved	-
0x00B0	CH0FTDA	Channel 0 First Tx Descr. Address
0x00B4	CH1FTDA	Channel 1 First Tx Descr. Address
0x00B8	CH2FTDA	Channel 2 First Tx Descr. Address
0x00BC	CH3FTDA	Channel 3 First Tx Descr. Address
0x00C0	reserved	-
0x00C4	reserved	-
0x00C8	CH0LRDA	Channel 0 Last Rx Descr. Address
0x00CC	CH1LRDA	Channel 1 Last Rx Descr. Address
0x00D0	CH2LRDA	Channel 2 Last Rx Descr. Address
0x00D4	CH3LRDA	Channel 3 Last Rx Descr. Address
0x00D8	reserved	-
0x00DC	reserved	-
0x00E0	CH0LTDA	Channel 0 Last Tx Descr. Address
0x00E4	CH1LTDA	Channel 1 Last Tx Descr. Address
0x00E8	CH2LTDA	Channel 2 Last Tx Descr. Address
0x00EC	CH3LTDA	Channel 3 Last Tx Descr. Address
0x00F0	VR	Version Register
0x00F4	ISPR	ISP Register
0x00F8	GCTLR	Global Control Register
0x00FC...0x00FF	reserved	-

Figure 6-2 : Global Registers

## 6.3 SCC Registers Overview

Each SCC register space (see table “Local Register Space”) contains the following registers:

The SCC registers are used to configure and control each of the four Serial Communication Controllers (SCC). The complete SCC register set exists four times, i.e. one for each SCC, distinguished by a SCC specific offset address.

The full 32 bit address location of each SCC register consists of:

- Base Address Register 0 (PCI Configuration Space, address location 0x10)
- SCC specific offset address:

SCC0: 0x0100

SCC1: 0x0180

SCC2: 0x0200

SCC3: 0x0280

- Register address offset, which is in the range 0x00 ...0x58

Most registers and register bit positions are shared by all SCC protocol modes (HDLC, ASYNC). Nevertheless the meaning (and name) of single bit positions might defer between different protocol modes. All registers are 32-bit organized registers.

Offset to Base Address 0x0100 (SCC0) 0x0180 (SCC1) 0x0200 (SCC2) 0x0280 (SCC3)	Register Name	
0x00	CMDR	Command Register
0x04	STAR	Status Register
0x08	CCR0	Channel Configuration Register 0
0x0C	CCR1	Channel Configuration Register 1
0x10	CCR2	Channel Configuration Register 2
0x14...0x2B	reserved	-
0x2C	BRR	Baud Rate Register
0x30...0x47	reserved	-
0x48	TCR	Termination Character Register
0x4C...0x53	reserved	-
0x54	IMR	Interrupt Mask Register
0x58	ISR	Interrupt Status Register
0x5C	ACR	Additional Configuration Register
0x60...0x7F	reserved	-

Figure 6-3 : SCC Registers

## 6.4 Global Registers

Basic function of register bits is the same as for the high speed serial controller used on the TPMC862.

### 6.4.1 GCMDR – Global Command Register (0x0000)

Bit	Symbol	Description	Access	Reset Value
31	CFGIQSCC3RX	<p>Configure Interrupt Queue SCC3 Receive Only valid, if action request bit 'AR' is set. The DMA (interrupt) controller will transfer interrupt vectors generated by the dedicated SCC receiver (3..0) to the corresponding interrupt queue which must be configured via 'CFGIQSCCiRX' command bits:</p> <p>'0': The DMA (interrupt) controller does NOT configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'.</p> <p>'1': Causes the DMA (interrupt) controller to configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'. On action request, the DMA (interrupt) controller will evaluate the corresponding interrupt queue base address and length registers which must have been programmed by software before.</p>	R/W	0
30	CFGIQSCC2RX	Configure Interrupt Queue SCC2 Receive (see above)	R/W	0
29	CFGIQSCC1RX	Configure Interrupt Queue SCC1 Receive (see above)	R/W	0
28	CFGIQSCC0RX	Configure Interrupt Queue SCC0 Receive (see above)	R/W	0

Bit	Symbol	Description	Access	Reset Value
27	CFGIQSCC3TX	<p>Configure Interrupt Queue SCC3 Transmit</p> <p>Only valid, if action request bit 'AR' is set.</p> <p>The DMA (interrupt) controller will transfer interrupt vectors generated by the dedicated SCC transmitter (3..0) to the corresponding interrupt queue which must be configured via 'CFGIQSCCiTX' command bits:</p> <p>'0': The DMA (interrupt) controller does NOT configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'.</p> <p>'1': Causes the DMA (interrupt) controller to configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'.</p> <p>On action request, the DMA (interrupt) controller will evaluate the corresponding interrupt queue base address and length registers which must have been programmed by software before.</p>	R/W	0
26	CFGIQSCC2TX	Configure Interrupt Queue SCC2 Transmit (see above)	R/W	0
25	CFGIQSCC1TX	Configure Interrupt Queue SCC1 Transmit (see above)	R/W	0
24	CFGIQSCC0TX	Configure Interrupt Queue SCC0 Transmit (see above)	R/W	0
23:22	-	Reserved (0 for reads)	R	0
21	CFGIQCFG	<p>Configure Interrupt Queue Configuration</p> <p>Only valid, if action request bit 'AR' is set.</p> <p>The DMA (interrupt) controller will transfer action request acknowledge/failure interrupt vectors to the configuration interrupt queue which must be configured via 'CFGIQCFG' command bits:</p> <p>'0': The DMA (interrupt) controller does NOT configure/re-configure the configuration interrupt queue, if action request bit 'AR' is set to '1'.</p> <p>'1': Causes the DMA (interrupt) controller to configure the configuration interrupt queue, if action request bit 'AR' is set to '1'.</p> <p>On action request, the DMA (interrupt) controller will evaluate the configuration interrupt queue base address and length registers which must have been programmed by software before.</p>	R/W	0
20:14	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value
13	TXPR3	<p>Transmit Poll Request Channel 3</p> <p>Self-clearing command bit, only valid in 'HOLD' bit controlled DMA controller mode (bit CMODE = '0' in register GMODE):</p> <p>'0': No Transmit Poll Request is performed. The corresponding DMA controller transmit channel is stopped when HOLD='1' has been detected in the current transmit descriptor.</p> <p>'1': Setting this bit to '1', when HOLD='1' has been detected in the current transmit descriptor, will cause the controller to poll the 'HOLD' bit in the current transmit descriptor, i.e. the controller reads the configuration word (DWORD 0) and next descriptor address (DWORD 1) of the current descriptor again. If the 'HOLD' bit is detected cleared ('0'), the DMA controller will branch to the next descriptor.</p> <p>When the DMA controller is not in 'HOLD' state, this command is discarded.</p>	R/W	0
12	TXPR2	Transmit Poll Request Channel 2	R/W	0
11	TXPR1	Transmit Poll Request Channel 1	R/W	0
10	TXPR0	Transmit Poll Request Channel 0	R/W	0
9	IMAR	Interrupt Mask Action Request	R/W	1
8:1	-	Reserved (0 for reads)	R	0
0	AR	Action Request	R/W	0

Figure 6-4 : Global Command Register

## 6.4.2 GSTAR – Global Status Register (0x0004)

Written by DMA Controller as interrupt indication

The Global Status Register indicates whether an action request was executed successfully or not. It also gives information about the interrupt source and which interrupt queue has been written to when INTA# is activated.

Nine interrupt queues are provided:

- four queues for receive interrupt vectors of the SCCs (SCCi, i=0...3)
- four queues for transmit interrupt vectors of the SCCs (SCCi, i=0...3)
- one queue for configuration interrupt vectors (action request acknowledge/failed)

To clear any bit in the status register, the host CPU must set the corresponding bit to '1' by register write access. Signal INTA# will be deasserted by the TPMC863 if all GSTAR Interrupt Indication bits are cleared.

Bit	Symbol	Description	Access	Reset Value
31	IISCC3RX	Interrupt Indication Queue SCC3 Receive	R/C	0
30	IISCC2RX	Interrupt Indication Queue SCC2 Receive	R/C	0
29	IISCC1RX	Interrupt Indication Queue SCC1 Receive	R/C	0
28	IISCC0RX	Interrupt Indication Queue SCC0 Receive These bits indicate whether at least one new interrupt vector was transferred into the corresponding receive interrupt queue: '0': No new interrupt vector was transferred into the corresponding queue. '1': At least one new interrupt vector was transferred into the corresponding queue.	R/C	0
27	IISCC3TX	Interrupt Indication Queue SCC3 Transmit	R/C	0
26	IISCC2TX	Interrupt Indication Queue SCC2 Transmit	R/C	0
25	IISCC1TX	Interrupt Indication Queue SCC1 Transmit	R/C	0
24	IISCC0TX	Interrupt Indication Queue SCC0 Transmit These bits indicate whether at least one new interrupt vector was transferred into the corresponding transmit interrupt queue: '0': No new interrupt vector was transferred into the corresponding queue. '1': At least one new interrupt vector was transferred into the corresponding queue.	R/C	0
23:22	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value
21	IICFG	Interrupt Indication Configuration Queue These bits indicate whether at least one new interrupt vector was transferred into the configuration interrupt queue:  '0': No new interrupt vector was transferred into the corresponding queue.  '1': At least one new interrupt vector was transferred into the corresponding queue.	R/C	0
20:2	-	Reserved (0 for reads)	R	0
<i>Action Request Result Status</i>				
1	ARF	Action Request Failed Status This bit indicates that an action request command was completed with an 'action request failed' condition:  '0': No action request was performed or no 'action request failed' condition occurred completing an action request.  '1': The last action request command was completed with an 'action request failed' condition.	R/C	0
0	ARACK	Action Request Acknowledge Status This bit indicates that an action request command was completed successfully:  '0': No action request was performed or completed successfully.  '1': The last action request command was completed successfully.	R/C	0

Figure 6-5 : Global Status Register

### 6.4.3 GMODE – Global Mode Register (0x0008)

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved (0 for reads)	R	0
0	CMODE	DMA Control Mode  '0' ' HOLD' bit control mode. The descriptor chain end condition is controlled via the 'HOLD' bit in each receive/transmit descriptor.  '1' Last Receive/Transmit Descriptor Address mode. The descriptor chain end condition is controlled via registers LRDA/LTDA.	R/W	0

Figure 6-6 : Global Mode Register

#### 6.4.4 IQLENR0 – Interrupt Queue Length Register 0 (0x000C)

Bit	Symbol	Description	Access	Reset Value
31:28	IQSCC0RXLEN	Interrupt Queue SCC3 Receive Length	R/W	0
27:24	IQSCC1RXLEN	Interrupt Queue SCC2 Receive Length	R/W	0
23:20	IQSCC2RXLEN	Interrupt Queue SCC1 Receive Length	R/W	0
19:16	IQSCC3RXLEN	Interrupt Queue SCC0 Receive Length  These bit fields determine the length of the corresponding receive interrupt queue (related to the respective SCC receive channel):  Queue Length = (1 + 'IQSCCiRXLEN') * 32 DWORDS 'IQSCCiRXLEN' = 0...15	R/W	0
15:12	IQSCC0TXLEN	Interrupt Queue SCC3 Transmit Length	R/W	0
11:8	IQSCC1TXLEN	Interrupt Queue SCC2 Transmit Length	R/W	0
7:4	IQSCC2TXLEN	Interrupt Queue SCC1 Transmit Length	R/W	0
3:0	IQSCC3TXLEN	Interrupt Queue SCC0 Transmit Length  These bit fields determine the length of the corresponding transmit interrupt queue (related to the respective SCC transmit channel):  Queue Length = (1 + 'IQSCCiTXLEN') * 32 DWORDS, 'IQSCCiTXLEN' = 0...15	R/W	0

Figure 6-7 : Interrupt Queue Length Register 0

#### 6.4.5 IQLENR1 – Interrupt Queue Length Register 1 (0x0010)

Bit	Symbol	Description	Access	Reset Value
31:24	-	Reserved (0 for reads)	R	0
23:20	IQCFGLEN	Interrupt Queue Configuration Length  Queue Length = (1 + 'IQCFGLEN') * 32 DWORDS, 'IQCFGLEN' = 0...15	R/W	0
19:0	-	Reserved (0 for reads)	R	0

Figure 6-8 : Interrupt Queue Length Register 1

#### 6.4.6 IQSCCiRXBAR – Interrupt Queue SCCi Receiver Base Address Register (i=0...3) (0x0014, 0x0018, 0x001C, 0x0020)

Bit	Symbol	Description	Access	Reset Value
31:2	IQSCCiRXBAR (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-9 : IQSCCiRXBAR Register

#### 6.4.7 IQSCCiTXBAR – Interrupt Queue SCCi Transmitter Base Address Register (i=0...3) (0x0024, 0x0028, 0x002C, 0x0030)

Bit	Symbol	Description	Access	Reset Value
31:2	IQSCCiTXBAR (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-10: IQSCCiTXBAR Register

#### 6.4.8 FIFO CR4 – FIFO Control Register 4 (0x0034)

Bit	Symbol	Description	Access	Reset Value
31:24	TFFTHRES3	Transmit FIFO Forward Threshold Channel 3	R/W	0
23:16	TFFTHRES2	Transmit FIFO Forward Threshold Channel 2	R/W	0
15:8	TFFTHRES1	Transmit FIFO Forward Threshold Channel 1	R/W	0

Bit	Symbol	Description	Access	Reset Value
7:0	TFFTHRES0	<p>Transmit FIFO Forward Threshold Channel 0</p> <p>These bit fields determine the channel specific Transmit FIFO Forward Threshold for the corresponding channel <i>i</i> in number of DWORDs. This threshold controls DMAC operation towards the serial channels (SCCi).</p> <p>As soon as the number of valid data belonging to a new frame in the central transmit FIFO is greater than the threshold, the DMAC will provide transmit data to the corresponding SCC. Once having started one frame, the DMAC will ignore this threshold providing all available data of the current frame to the SCC. Threshold operation starts again with the beginning of a new frame. Frames shorter than the threshold will be transferred as soon as a frame end indication is detected by the DMAC.</p> <p><i>Note: Programming TFFTHRESi to zero will disable the threshold causing the DMAC to transfer all data immediately. This may be useful for not frame oriented data transmission, e.g. in ASYNC protocol mode.</i></p>	R/W	0

Figure 6-11 : FIFO Control Register 4

#### 6.4.9 IQCFGBAR – Interrupt Queue Configuration Base Address Register (0x003C)

Bit	Symbol	Description	Access	Reset Value
31:2	IQCFGBAR (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-12: IQCFGBAR Register

#### 6.4.10 CHiCFG – Channel *i* Configuration Register (*i*=0...3) (0x0050, 0x005C, 0x0068, 0x0074)

Bit	Symbol	Description	Access	Reset Value
31:28	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value
27	MRFI	Mask Receive FI Interrupt (Channel i) This bit enables/disables the receive FI interrupt indication for the DMA channel, the register is dedicated to (i=3..0):  '0': FI interrupt generation is enabled for the dedicated DMA receive channel.  '1': FI interrupt generation is disabled for the dedicated DMA receive channel.	R/W	0
26	MTFI	Mask Transmit FI Interrupt (Channel i) This bit enables/disables the transmit FI interrupt indication for the DMA channel, the register is dedicated to (i=3..0):  '0': FI interrupt generation is enabled for the dedicated DMA transmit channel.  '1': FI interrupt generation is disabled for the dedicated DMA transmit channel.	R/W	0
25	MRERR	Mask Receive ERR Interrupt (Channel i) This bit enables/disables the receive ERR interrupt indication for the DMA channel, the register is dedicated to (i=3..0):  '0': ERR interrupt generation is enabled for the dedicated DMA receive channel.  '1': ERR interrupt generation is disabled for the dedicated DMA receive channel.	R/W	0
24	MTERR	Mask Transmit ERR Interrupt (Channel i) This bit enables/disables the transmit ERR interrupt indication for the DMA channel, the register is dedicated to (i=3..0):  '0': ERR interrupt generation is enabled for the dedicated DMA transmit channel.  '1': ERR interrupt generation is disabled for the dedicated DMA transmit channel.	R/W	0
23	-	Reserved (0 for reads)	R	0
22	RDR	Reset DMA Receiver (Channel i) Self-clearing command bit. This command resets the specific DMA controller receive channel and also flushes the receive data FIFO. After reset, the respective DMA channel is in its initial state equal to the reset state after power on. The receive data FIFO will not accept any receive data until the IDR command is successfully finished.	R/W	0

Bit	Symbol	Description	Access	Reset Value
21	RDT	Reset DMA Transmitter (Channel i) Self-clearing command bit. This command resets the specific DMA controller transmit channel. After reset, the respective DMA channel is in its initial state equal to the reset state after power on.	R/W	0
20	IDR	Initialize DMA Receiver (Channel i) Self-clearing command bit. This command causes the specific DMA receive channel to fetch the base descriptor address from register CHiBRDA and to branch to the corresponding descriptor. Afterwards normal DMA operation on the receive descriptor list is performed depending on the selected DMA control mode.  <i>Note: To avoid unexpected DMA controller behavior, it is recommended to apply 'IDR' command only, if the specific DMA channel is in reset state.</i>	R/W	0
19	IDT	Initialize DMA Transmitter (Channel i) Self-clearing command bit. This command causes the specific DMA transmit channel to fetch the base descriptor address from register CHiBTDA and to branch to the corresponding descriptor. Afterwards normal DMA operation on the transmit descriptor list is performed depending on the selected DMA control mode.  <i>Note: To avoid unexpected DMA controller behavior, it is recommended to apply 'IDT' command only, if the specific DMA channel is in reset state.</i>	R/W	0
18:0	-	Reserved (0 for reads)	R	0

Figure 6-13: CHiCFG Register

#### 6.4.11 CHiBRDA – Channel i Base Receive Descriptor Address Register (i=0...3) (0x0054, 0x0060, 0x006C, 0x0078)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiBRDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-14: CHiBRDA Register

### 6.4.12 CHiBTDA – Channel i Base Transmit Descriptor Address Register (i=0...3) (0x0058, 0x0064, 0x0070, 0x007C)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiBTDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-15: CHiBTDA Register

### 6.4.13 CHiFRDA – Channel i First (Current) Receive Descriptor Address Register (i=0...3) (0x0098, 0x009C, 0x00A0, 0x00A4)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiFRDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-16: CHiFRDA Register

### 6.4.14 CHiFTDA – Channel i First (Current) Transmit Descriptor Address Register (i=0...3) (0x00B0, 0x00B4, 0x00B8, 0x00BC)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiFTDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-17: CHiFTDA Register

### 6.4.15 CHiLRDA – Channel i Last Receive Descriptor Address Register (i=0...3) (0x00C8, 0x00CC, 0x00D0, 0x00D4)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiLRDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-18: CHiLRDA Register

### 6.4.16 CHiLTDA – Channel i Last Transmit Descriptor Address Register (i=0...3) (0x00E0, 0x00E4, 0x00E8, 0x00EC)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiLTDA (31:2)		R/W	0
1:0	-	Reserved (0 for reads)	R	0

Figure 6-19: CHiLTDA Register

### 6.4.17 VR – Version Register (0x00F0)

Bit	Symbol	Description	Access	Reset Value
31:16	SPC_ID	Reserved (0 for reads) or special customer ID register	R	0
15:0	VER	Revision of FPGA Design	R	0x0001

Figure 6-20: Version Register

### 6.4.18 ISPR – In-System-Programming Register (0x00F4)

This register is reserved for (factory) reprogramming of the FPGA configuration flash.

No write accesses shall be done to this address, as permanent damage may occur.

### 6.4.19 GCTLR – Global Control Register (0x00F8)

Bit	Symbol	Description	Access	Reset Value
31:20	-	Reserved (0 for reads)	R	0
19	EEDO	Serial EEPROM Data Out (Q)	R	-
18	EEDI	Serial EEPROM Data In (D)	R/W	0
17	EECS	Serial EEPROM Chip Select (S)	R/W	0
16	EESK	Serial EEPROM Clock (C)	R/W	0
15:1	-	Reserved	R	-
0	LRST	Local Reset Self-clearing command bit. The complete local part of the device is reset. Only the registers in the PCI configuration space keep their values.	R/W	0

Figure 6-21: Global Control Register

## 6.5 SCC Channel Specific Registers

### 6.5.1 CMDR – Command Register (0x0100, 0x0180, 0x0200, 0x0280)

Bit	Symbol	Description	Access	Reset Value
31:25	-	Reserved (0 for reads)	R	0
24	XRES	<p>Transmitter Reset (Self-clearing)</p> <p>'1': The transmit FIFOs (Main and SCC FIFO) are cleared and the transmitter protocol engines are reset to their initial state. A transmitter reset command is recommended after all changes in protocol mode configurations (e.g. switching between the protocol engines HDLC/ASYNC or sub-modes of HDLC).</p> <p><i>Note: A transmit clock must be present.</i></p>	R/W	0
23:17	-	Reserved (0 for reads)	R	0
16	RRES	<p>Receiver Reset (Self-clearing)</p> <p>'1': The receive SCC FIFO is flushed and the receiver protocol engine is reset. Recommended after changes in protocol configuration (switching between the protocol engines or sub-modes of HDLC).</p> <p><i>Note: A receive clock must be present.</i></p>	R/W	0
15:0	-	Reserved (0 for reads)	R	0

Figure 6-22: Command Register

### 6.5.2 STAR – Status Register (0x0104, 0x0184, 0x0204, 0x0284)

Bit	Symbol	Description	Access	Reset Value
31:25	-	Reserved (0 for reads)	R	0
24	CTS	<p>Clear To Send Input Signal State</p> <p>'0': CTS# input signal is inactive (high level) '1': CTS# input signal is active (low level)</p> <p><i>Note: A transmit clock must be present. Optionally this input can be programmed to generate an interrupt on signal level changes.</i></p>	R	-
23:22	-	Reserved (0 for reads)	R	0

21	CD	CD (Carrier Detect) Input Signal State '0': CD input signal is inactive (low level) '1': CD input signal is active (high level) <i>Note: A receive clock must be present. Optionally this input can be programmed to generate an interrupt on signal level changes.</i>	R	-
20	-	Reserved (0 for reads)	R	0
19	DPLA	DPLL Asynchronous This bit is only valid if the receive clock is recovered by the DPLL and FM0, FM1 or Manchester data encoding is selected. It is set when the DPLL has lost synchronization. In this case reception is disabled until synchronization has been regained. In addition transmission is interrupted in all cases where transmit clock is derived from the DPLL. '0' DPLL is synchronized. '1' DPLL is asynchronous (re-synchronization process is started automatically).	R	-
18:1	-	Reserved (0 for reads)	R	0
0	DSR3	Data Set Ready Channel 3 <b>(Only on channel 3!)</b>	R	-

Figure 6-23: Status Register

### 6.5.3 CCR0 – Channel Configuration Register 0 (0x0108, 0x0188, 0x0208, 0x0288)

Bit	Symbol	Description	Access	Reset Value
31:23	-	Reserved (0 for reads)	R	0
22:20	SC	Serial Port Configuration '000': NRZ data encoding '010': NRZI data encoding '100': FM0 data encoding '101': FM1 data encoding '110': Manchester data encoding others: reserved	R/W	000
19:18	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value
17:16	SM	Serial Port Mode Selects the protocol engine: '00': HDLC synchronous '01': reserved '10': reserved '11': Asynchronous	R/W	00
15:13	-	Reserved (0 for reads)	R	0
12	VIS	Masked Interrupts Visible '0': Masked interrupt status bits are not visible on interrupt status register (ISR) read accesses. '1': Masked interrupt status bits are visible in the ISR. To clear these interrupt flags, the host CPU must write '1' to the corresponding ISR bit. <i>Note: Masked interrupts will not generate an interrupt vector to the interrupt controller.</i>	R/W	0
11:8	-	Reserved (0 for reads)	R	0
7	BCR	Bit Clock Rate Only valid in asynchronous mode. '0': Isochronous (Bit Clock Rate x1). Bits are sampled once. '1': Standard asynchronous (Bit Clock Rate x16). Bits are sampled 16 times. The result is determined by a majority decision of 3 samples around the bit center. NRZ encoding has to be selected.	R/W	0
6	-	Reserved (0 for reads)	R	0
5	TOE	Transmit Clock Out Enable '0': TxC is input (DCE mode) '1': TxC is output (DTE#, if <b>not</b> used as TxC input, i.e. ACR.TCS is not '100') <i>Note: The Transceiver direction of TxC is set according to TOE.</i>	R/W	0
4:0	-	Reserved (0 for reads)	R	0

Figure 6-24: Channel Configuration Register 0

#### 6.5.4 CCR1 – Channel Configuration Register 1 (0x010C, 0x018C, 0x020C, 0x028C)

Bit	Symbol	Description	Access	Reset Value
31:21	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value															
20	RTS	Request To Send pin control (async./isochr.) The request to send pin RTS# can be controlled as an output autonomously or via setting/clearing bit 'RTS'. '0': Pin RTS# (output) pin is controlled autonomously. The behavior of this pin depends on bit 'FRTS'. '1': Pin RTS# can be controlled by software. The output level of this pin depends on bit 'FRTS'.	R/W	0															
19	FRTS	Flow Control (RTS) Function of RTS# depends on 'RTS' and 'FRTS'. <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>FRTS</th> <th>RTS</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Pin RTS# is controlled autonomously. RTS is asserted (low) when data is available in the transmit FIFO.</td> </tr> <tr> <td>1</td> <td>0</td> <td>Pin RTS# is controlled autonomously. RTS is asserted (low) if the SCC receive FIFO is nearly empty (2 bytes) and de-asserted (high) if the receive FIFO fill level reaches 12 bytes.</td> </tr> <tr> <td>0</td> <td>1</td> <td>Forces RTS# to low (asserted).</td> </tr> <tr> <td>1</td> <td>1</td> <td>Forces RTS# to high (de-asserted).</td> </tr> </tbody> </table> <p><i>Note: A transmit clock is necessary. In HDLC mode the RTS pin is always controlled by software.</i></p>	FRTS	RTS		0	0	Pin RTS# is controlled autonomously. RTS is asserted (low) when data is available in the transmit FIFO.	1	0	Pin RTS# is controlled autonomously. RTS is asserted (low) if the SCC receive FIFO is nearly empty (2 bytes) and de-asserted (high) if the receive FIFO fill level reaches 12 bytes.	0	1	Forces RTS# to low (asserted).	1	1	Forces RTS# to high (de-asserted).	R/W	0
FRTS	RTS																		
0	0	Pin RTS# is controlled autonomously. RTS is asserted (low) when data is available in the transmit FIFO.																	
1	0	Pin RTS# is controlled autonomously. RTS is asserted (low) if the SCC receive FIFO is nearly empty (2 bytes) and de-asserted (high) if the receive FIFO fill level reaches 12 bytes.																	
0	1	Forces RTS# to low (asserted).																	
1	1	Forces RTS# to high (de-asserted).																	
18	FCTS	Flow Control (CTS) (async./isochr.) '0': Transmitter is stopped, if CTS# input signal is inactive (high) and enabled if asserted (low). In ASYNC mode, the current byte is completely sent even if CTS# becomes inactive during transmission. '1': Transmitter is always enabled. <i>Note: In ASYNC mode the current byte is completely sent, even if CTS# becomes deasserted during transmission.</i>	R/W	0															
17:16	-	Reserved (0 for reads)	R	0															
15:14	MDS	Mode Select (hdlc) Selects the HDLC sub mode. '00': reserved '01': reserved '10': Address Mode 0 '11': Extended transparent mode (bit transparent transmission/reception)	R/W	00															
13:9	-	Reserved (0 for reads)	R	0															

Bit	Symbol	Description	Access	Reset Value
8	TLP	<p>Test Loop</p> <p>The test loop is closed at the far end of serial transmit and receive line just before the respective TxD and RxD pins:</p> <p>'0': Test loop disabled.</p> <p>'1': Test loop enabled.</p>	R/W	0
7	TOIE	<p>Time Out Indication Enable (async./isochr.)</p> <p>A 'block end' indication is inserted in the receive FIFO if a time out occurs. The current receive descriptor will be finished.</p> <p>'0': Time Out function disabled.</p> <p>'1': Time Out function enabled:</p> <p><i>Note: A time out event will generate a 'TIME' interrupt (if unmasked).</i></p>	R/W	0
6:0 (asyn)	TOLEN	<p>Time Out Length (async./isochr.)</p> <p>Determines the time out period. If there is no receive line activity for the configured period of time, a time out indication is generated if enabled via bit 'TOIE'.</p> <p>The period of time is programmable in multiples of <b>a single</b> character frame length (CFL) time equivalents including start, parity and stop bits:</p> $T_{\text{TOUIT}} = (\text{TOLEN} + 1) * 1 * \text{CFL}$	R/W	0
1 (hdlc)	CRL	<p>CRC Reset Value (hdlc)</p> <p>Defines the initial value of the CRC generators:</p> <p>'0': Initial value is 0xFFFF (16 bit CRC), 0xFFFFFFFF (32 bit CRC); (default value for most HDLC applications)</p> <p>'1': Initial value is 0x0000 (16 bit CRC), 0x00000000 (32 bit CRC).</p>	R/W	0
0 (hdlc)	C32	<p>CRC-32 Select (hdlc)</p> <p>This bit enables 32-bit CRC operation for transmit and receive.</p> <p>'0': 16-bit CRC generation/checking.</p> <p>'1': 32-bit CRC generation/checking.</p>	R/W	0

Figure 6-25: Channel Configuration Register 1

## 6.5.5 CCR2 – Channel Configuration Register 2 (0x0110, 0x0190, 0x0210, 0x0290)

Bit	Symbol	Description	Access	Reset Value
31:30	-	Reserved (0 for reads)	R	0
29:28 (asyn.)	CHL	Character Length (asyn./isochr.) '00': 8 bit data '01': 7 bit data '10': 6 bit data '11': 5 bit data	R/W	00
27	RAC	Receiver active '0': Receiver inactive, receive line is ignored. '1': Receiver active.	R/W	0
26:25	-	Reserved (0 for reads)	R	0
24 (asyn.)	STOP	Stop Bit Number (asyn./isochr.) '0': 1 stop bit per character '1': 2 stop bits per character	R/W	0
23:22 (asyn.)	PAR	Parity Format (asyn./isochr.) '00': Space ('0') is inserted as parity bit '01': Odd parity '10': Even parity '11': Mark ('1') is inserted as parity bit <i>Note: The received parity bit (and parity error) is stored in the receive buffer if bit 'RFDF' = '1'.</i>	R/W	0
21 (asyn.)	PARE	Parity Enable (asyn./isochr.) '0': Parity generation/checking is disabled. '1': Parity generation/checking is enabled.	R/W	0
24:23 (hdlc)	-	Reserved (0 for reads)	R	0
22 (hdlc)	DRCRC	Disable Receive CRC Checking (hdlc) '0': The receiver expects a 16 or 32 bit CRC within a HDLC frame. '1': The receiver does not expect a CRC. <i>Note: A received checksum (2 or 4 bytes) is always forwarded to the receive buffer as data.</i>	R/W	0
21 (hdlc)	-	Reserved (0 for reads)	R	0
20	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value										
19	RFDF	Receive FIFO Data Format (async./isochr.) '0': No additional status information stored. '1': Status byte is stored <b>after each data byte</b> to the receive buffer:  <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td style="text-align: center;">15</td> <td style="text-align: center;">14</td> <td style="text-align: center;">13...9</td> <td style="text-align: center;">8</td> <td style="text-align: center;">7.....0</td> </tr> <tr> <td style="text-align: center;">parity error</td> <td style="text-align: center;">frame error</td> <td style="text-align: center;">reserved</td> <td style="text-align: center;">parity bit</td> <td style="text-align: center;">data byte</td> </tr> </table> <i>Note: RFDF value is only evaluated while Receiver Reset 'RRES' is active.</i>	15	14	13...9	8	7.....0	parity error	frame error	reserved	parity bit	data byte	R/W	0
15	14	13...9	8	7.....0										
parity error	frame error	reserved	parity bit	data byte										
18	RFTH	Receive FIFO Threshold '0': Maximum PCI bus write burst length for receive data is set to 15 DWORDs. '1': Maximum PCI bus write burst length for receive data is set to 1 DWORD.	R/W	0										
17:1	-	Reserved (0 for reads)	R	0										
0 (hdlc)	XCRC	Transmit CRC Checking Mode (hdlc) '0': The transmit checksum (2 or 4 bytes) is generated and appended to the transmit data. '1': The transmit checksum is not generated.	R/W	0										

Figure 6-26: Channel Configuration Register 2

### 6.5.6 BRR – Baud Rate Register (0x012C, 0x01AC, 0x022C, 0x02AC)

Bit	Symbol	Description	Access	Reset Value
31:12	-	Reserved (0 for reads)	R	0
11:8	BRM	BRM(3:0) Baud Rate Factor 'M'	R/W	0
7:6	-	Reserved (0 for reads)	R	0
5:0	BRN	BRN(5:0) Baud Rate Factor 'N'	R/W	0

Figure 6-27: Baud Rate Register

These values determine the division factor of the baud rate generator. The baud rate generator input clock depends on the selected clock source. (see also chapters “Clock Sources” and “Additional Configuration Register”)

The division factor k is calculated by:

$$k = (N + 1) \times 2^M$$

with M=0..15 and N=0..63.

The resulting output frequency of the baud rate generator is:

$$f_{BRG} = f_{in} / k$$

### 6.5.7 TCR – Termination Character Register (0x0148, 0x01C8, 0x0248, 0x02C8)

Bit	Symbol	Description	Access	Reset Value
31:16	-	Reserved (0 for reads)	R	0
15	TCDE	Termination Character Detection Enable (async./isochr.) '0': No receive termination character detection '1': Termination character detection is enabled. The receive data is analyzed for the termination character TC. When character is detected, a 'block end' and a 'TCDI' interrupt (if enabled) is generated.	R/W	0
14:8	-	Reserved (0 for reads)	R	0
7:0	TC	Termination Character (async./isochr.) Defines the termination character which is monitored on the receive data stream if enabled via bit 'TCDE'.	R/W	0

Figure 6-28: Termination Character Register

## 6.5.8 IMR – Interrupt Mask Register (0x0154, 0x01D4, 0x0254, 0x02D4)

Bit	Symbol	Description	Access	Reset Value
31:19	-	'0': interrupt is NOT masked, interrupt via INTA# IS generated '1': interrupt is masked, NO interrupt on INTA# generated	R	1
18	ALLS		R/W	1
17	-		R	1
16 (hdlc)	XDU		R/W	1
16 (asyn)	-		R	1
15	-		R	1
14	CSC		R/W	1
13:10	-		R	1
9 (asyn)	BRK		R/W	1
8 (asyn)	BRKT		R/W	1
7 (asyn)	TCD		R/W	1
6 (asyn)	TIME		R/W	1
5 (asyn)	PERR		R/W	1
4 (asyn)	FERR		R/W	1
9:4 (hdlc)	-		R	1
3	PLLA		R/W	1
2	CDSC		R/W	1
1	RFO		R/W	1
0	-		R	1

Figure 6-29: Interrupt Mask Register

**Unused interrupts shall be masked to avoid unwanted behavior.**

**Especially CSC and CDSC should be masked if CTS and CD inputs are unconnected.**

## 6.5.9 ISR – Interrupt Status Register (0x0158, 0x01D8, 0x0258, 0x02D8)

Bit	Symbol	Description	Access	Reset Value
31:19	-	Reserved (0 for reads)	R	0
18	ALLS	<p>ALL Sent Interrupt</p> <p>HDLC Mode: This bit is set to '1' if the last bit of the current HDLC frame is sent out via pin TxD,</p> <p>ASYNC/ISOCHR Mode: This bit is set to '1', if the last character is completely sent via pin TxD and no further data is stored in the SCC transmit FIFO, i.e. the transmit FIFO is empty.</p>	R/C	0
17	-	Reserved (0 for reads)	R	0
16 (hdlc)	XDU	<p>Transmit Data Underrun Interrupt (hdlc)</p> <p>HDLC Mode: This bit is set to '1', if the current frame was terminated by the SCC with an abort sequence, because neither a 'frame end / block end' indication was detected in the FIFO (to complete the current frame) nor more data is available in the SCC transmit FIFO.</p> <p><i>Note: The transmitter is stopped if this condition occurs and needs to be reset via command bit 'XRES' in register CMDR.</i></p>	R/C	0
16 (asyn)	-	Reserved (0 for reads)	R	0
15	-	Reserved (0 for reads)	R	0
14	CSC	<p>CTS# Status Change Interrupt</p> <p>This bit is set to '1', if a transition occurs on signal CTS#. The current state of signal CTS# is monitored by status bit 'CTS' in status register STAR.</p>	R/C	0
13:10	-	Reserved (0 for reads)	R	0
9 (asyn)	BRK	<p>Break Interrupt (asyn./isochr.)</p> <p>This bit is set to '1', if a break condition was detected on the receive line, i.e. a low level for a time equal to (character length + parity bit + stop bit(s)) bits depending on the selected ASYNC character format.</p>	R/C	0
8 (asyn)	BRKT	<p>Break Terminated Interrupt (asyn./isochr.)</p> <p>This bit is set to '1', if a previously detected break condition on the receive line is terminated by a low to high transition.</p>	R/C	0
7 (asyn)	TCD	<p>Termination Character Detected Interrupt (asyn./isochr.)</p> <p>This bit is set to '1', if a termination character is detected in the receive data stream. The SCC will insert a 'frame end / block end' indication to the SCC receive FIFO which causes the DMAC to finish the current receive descriptor.</p>	R/C	0

Bit	Symbol	Description	Access	Reset Value
6 (asyn)	TIME	Time Out Interrupt (asyn./isochr.)  This bit is set to '1', if the time out limit is exceeded, i.e. no new character was received in a programmable period of time (refer to register CCR1 bit fields 'TOIE' and 'TOLEN' for more information).	R/C	0
5 (asyn)	PERR	Parity Error Interrupt (asyn./isochr.)  This bit is only valid if parity checking/generation is enabled via bit 'PARE' in register CCR2. It is set to '1', if a character with wrong parity has been received. If enabled via bit 'RFDF', this error status is additionally stored in the receive status byte generated for each receive character.	R/C	0
4 (asyn)	FERR	Frame Error Interrupt (asyn./isochr.)  This bit is set to '1', if a character framing error is detected, i.e. a '0' was sampled at a position where a stop bit '1' was expected due to the selected character format.	R/C	0
9:4 (hdlc)	-	Reserved (0 for reads)	R	0
3	PLLA	DPLL Asynchronous Interrupt  This bit is only valid, if the receive clock is derived from the internal DPLL and FM0, FM1 or Manchester data encoding is selected (depending on the selected clock source and data encoding mode). It is set to '1' if the DPLL has lost synchronization. Reception is disabled until synchronization has been regained again. If the transmitter is supplied with a clock derived from the DPLL, transmission is also interrupted.	R/C	0
2	CDSC	Carrier Detect Status Change Interrupt  This bit is set to '1', if a state transition has been detected at signal CD. Because only a state transition is indicated via this interrupt, the current status can be evaluated by reading bit 'CD' in status register STAR.	R/C	0
1	RFO	Receive FIFO Overflow Interrupt  This bit is set to '1', if receive data got lost because of a SCC receive FIFO full condition.	R/C	0
0	-	Reserved (0 for reads)	R	0

Figure 6-30: Interrupt Status Register

**If CCR0.VIS is set to '1' then masked interrupt status bits will be visible in the ISR.**

**To clear these interrupt flags, the host CPU must write '1' to the corresponding ISR bit.**

### 6.5.10 ACR – Additional Configuration Register (0x015C, 0x01DC, 0x025C, 0x02DC)

Bit	Symbol	Description	Access	Reset Value
31:20	-	Reserved (0 for reads)	R	0
19	ETRBO	Extended Transparent mode Receive Bit Order  '0': Receive Data is LSB first (first received bit at position 0)  '1': MSB first (first received bit at position 7)  <i>Note: Setting this bit to '1' might be useful to detect special characters more easily by software in the receive data stream. After detection of the byte alignment, the bits in each byte have to be mirrored to get the original data bytes.</i>	R/W	0
18	DTR3	Data Terminal Ready Channel 3 <b>(Only on channel 3!)</b>	R/W	0
17	CDOUT	CD Output Value	R/W	0
16	CDDIR	CD Direction  '0': CD is input '1': CD is output	R/W	0
15	DCMRST	Reset the Clock Multiplier  '0': Clock Multiplier is running '1': Clock Multiplier is held in Reset  <i>Note: The Clock Multiplier should always be reset after changing its input clock</i>	R/W	0
14	X4MULT	Multiply BRG Input Clock x4  '0': No clock frequency change '1': BRG clock frequency is multiplied by 4	R/W	0
13	RXCINV	Invert RxC  '0': No inverting of RxC '1': RxC is inverted	R/W	0
12	TXCINV	Invert TxC  '0': No inverting of TxC '1': TxC is inverted	R/W	0
11:10	RCS	Receiver Clock Source  '00': BRG Output Clock '01': external RxC Input signal '10': DPLL Output Clock '11': reserved	R/W	00

Bit	Symbol	Description	Access	Reset Value
9:7	TCS	Transmitter Clock Source '000': BRG Output Clock '001': external RxC Input signal '010': external TxC (Input direction, CCR0.TOE=0) '011': DPLL Output Clock '100': BRG Output Clock divided by 16 others: reserved	R/W	000
6:4	BCS	BRG Clock Source '000': Oscillator 1 Clock 14.7456 MHz '001': Oscillator 2 Clock 24 MHz '010': Oscillator 3 Clock 10 MHz '011': external RxC Input signal '100': external TxC (Input direction, CCR0.TOE=0) others: reserved	R/W	000
3	RTSCLK	Enable TxC Output on RTS pin '0': No TxC Output on RTS (normal function) '1': TxC Output Enabled on RTS	R/W	0
2:0	MODE	Transceiver Mode (M2:M0, see following table)	R/W	111

Figure 6-31: Additional Configuration Register

**The input frequency range of the x4 clock multiplier is 4.5 MHz to 28 MHz, these values must never be exceeded to ensure proper function of the clock multiplier.**

Transceiver Mode	M2	M1	M0	Driver/Receiver Mode
Not Used (Default V.11)	0	0	0	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.11</b>
EIA-530A	0	0	1	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.11</b>
EIA-530	0	1	0	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.11</b>
X.21	0	1	1	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.11</b>
V.35	1	0	0	TxD,TxC,RxD,RxC: <b>V.35</b> / RTS,CTS,CD: <b>V.28</b>
EIA-449/V.36	1	0	1	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.11</b>
V.28/EIA-232	1	1	0	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>V.28</b>
No Cable (high impedance)	1	1	1	TxD,TxC,RxD,RxC,RTS,CTS,CD: <b>Z</b>

Figure 6-32: Physical Interface Mode Selection

# 7 Configuration Hints

## 7.1 Big / Little Endian

PCI – Bus (Little Endian):

Byte 0	AD[7..0]
Byte 1	AD[15..8]
Byte 2	AD[23..16]
Byte 3	AD[31..24]

**The TPMC863 expects all accesses by the host and all data structures in the host RAM to be 'Little Endian'.**

**Transmit data in lower bytes is sent first.**

**Receive data that was received earlier is stored at lower bytes.**

## 7.2 Configuration EEPROM

An industry standard M93C56 serial EEPROM with 2 Kbit is connected to the FPGA and can be accessed through the GCTLR register. The EEPROM is configured for word (16 bit) accesses.

The addresses 0x00..0x5F are factory programmed with configuration information used by the software drivers and **must not be overwritten**. The other addresses (0x60..0x7F) are user programmable.

The configuration EEPROM contains the following data:

- Vendor ID
- Vendor Device ID
- Subsystem Vendor ID
- Subsystem Device ID
- The module version and revision
- The oscillator frequencies in Hz
- The physical interface attached to the serial channels
- The maximal baud rate of the transceivers in bps
- The supported control signals of the serial channels

For the physical interfaces and the control signals applies: Bit 3 represents UART channel 3 and bit 0 represents UART channel 0. The appropriate bit is set to '1' for each UART channel attached to the physical interface represented by the word. Bit 15 to bit 4 are always '0'.

Address	Configuration Register	TPMC863
0x00	Vendor ID	0x1498
0x01	Device ID	0x035F
0x02	Subsystem Vendor ID	0x1498
0x03	Subsystem ID	s.b.
0x04	Module Version	0x0100
0x05	Module Revision	0x0000
0x06	Module Variant	s.b.
0x07	EEPROM Revision	0x0001
0x08	Oscillator 1 Frequency (high)	0x00E1
0x09	Oscillator 1 Frequency (low)	0x0000
0x0A	Oscillator 2 Frequency (high)	0x016E
0x0B	Oscillator 2 Frequency (low)	0x3600
0x0C	Oscillator 3 Frequency (high)	0x0098
0x0D	Oscillator 3 Frequency (low)	0x9680
0x0E-0x0F	Reserved	-
0x10	RS232 Channels	0x000F
0x11	RS422 Channels	0x000F
0x12-0x14	Reserved	-
0x15	Multiprotocol Channels	0x000F
0x16-0x1E	Reserved	-
0x1F	Programmable Interfaces	0x000F
0x20	Max Data Rate RS232 (high)	0x0001
0x21	Max Data Rate RS232 (low)	0xF400
0x22	Max Data Rate RS422 (high)	0x0098
0x23	Max Data Rate RS422 (low)	0x9680
0x24-0x2F	Reserved	-
0x30	RxD & TxD	0x000F
0x31	RTS & CTS & CD	0x000F
0x32	Full modem	0x0008
0x33-0x5F	Reserved	-
0x60-0x7F	<b>User programmable space</b>	-

Figure 7-1 : Configuration EEPROM Data

Subsystem ID:	TPMC863-10	0x000A
Module Variant:	TPMC863-10	0x000A

## 7.3 Additional Termination Resistors

The Rx-/Tx-Clock and Data lines are connected to LTC1546 transceivers with internal termination; the control lines (RTS, CTS, CD) are connected to LTC1544 without termination, as termination of these signals is not necessary in the normal case. If the application requires termination of these lines standard 120Ω resistors in 0603 package can be mounted on the provided pads (see following two figures for assignment and positions).

Signal	Resistor	Signal	Resistor
CTS0	R4	CTS2	R1
CD0	R5	CD2	R2
RTS0	R6	RTS2	R3
CTS1	R12	CTS3	R9
CD1	R13	CD3	R10
RTS1	R14	RTS3	R11

Figure 7-2: Termination Resistor Assignments

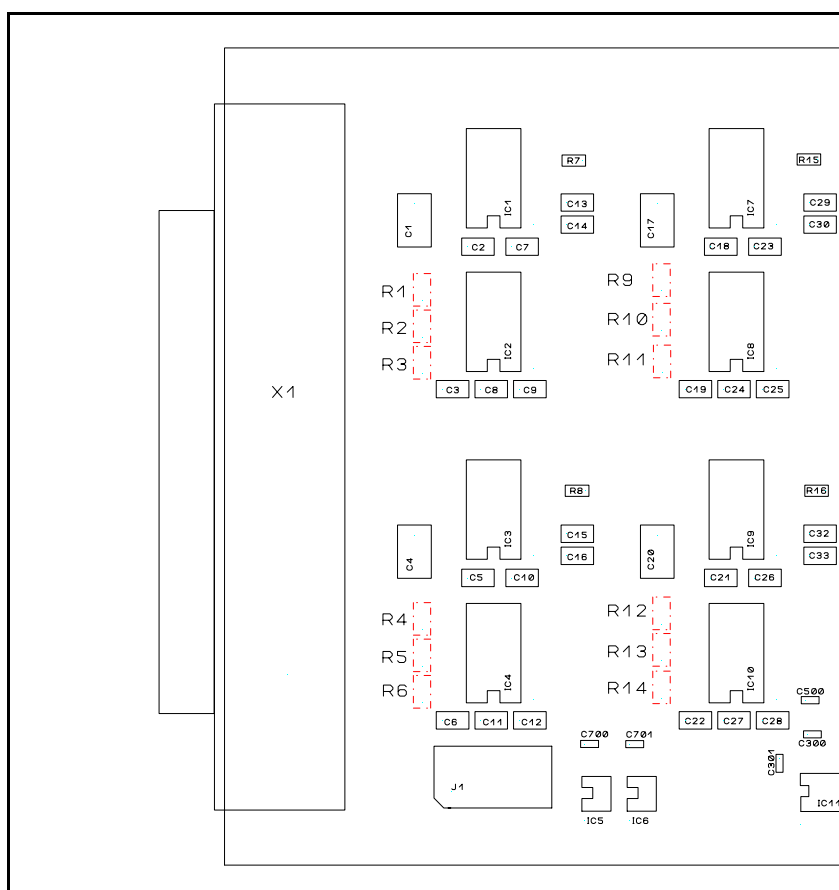


Figure 7-3: Termination Resistor Positions

## **8 Pin Assignment – I/O Connector**

### **8.1 Front Panel I/O Connector**

AMP 787082-7 or compatible

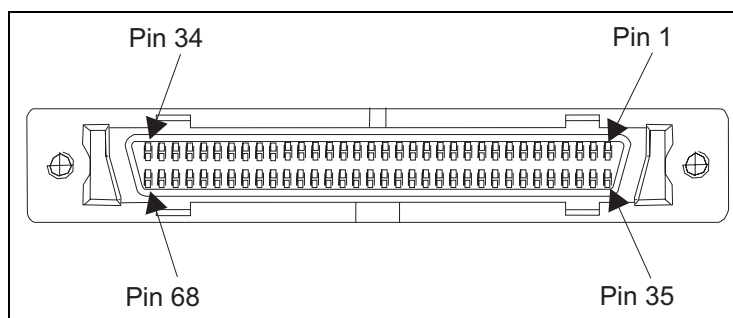


Figure 8-1 : Front Panel I/O Connector Numbering

Pin	Signal	Port	Pin	Signal	Port
1	CDA/-	0	35	CDA/-	1
2	CDB/+		36	CDB/+	
3	RXDA/-		37	RXDA/-	
4	RTSA/-		38	RTSA/-	
5	TXDA/-		39	TXDA/-	
6	CTSA/-		40	CTSA/-	
7	RTSB/+		41	RTSB/+	
8	CTSB/+		42	CTSB/+	
9	GND		43	GND	
10	TXDB/+		44	TXDB/+	
11	RXDB/+		45	RXDB/+	
12	TXCA/-		46	TXCA/-	
13	TXCB/+		47	TXCB/+	
14	GND		48	GND	
15	RXCA/-		49	RXCA/-	
16	RXCB/+		50	RXCB/+	
17	CDA/-	2	51	CDA/-	3
18	CDB/+		52	CDB/+	
19	RXDA/-		53	RXDA/-	
20	RTSA/-		54	RTSA/-	
21	TXDA/-		55	TXDA/-	
22	CTSA/-		56	CTSA/-	
23	RTSB/+		57	RTSB/+	
24	CTSB/+		58	CTSB/+	
25	GND		59	GND	
26	TXDB/+		60	TXDB/+	
27	RXDB/+		61	RXDB/+	
28	TXCA/-		62	TXCA/-	
29	TXCB/+		63	TXCB/+	
30	GND		64	GND	
31	RXCA/-		65	RXCA/-	
32	RXCB/+		66	RXCB/+	
33	DSRB/+	3	67	DTRB/+	
34	DSRA/-		68	DTRA/-	

Figure 8-2 : Front I/O Pin Assignment

**In V.28 (single-ended) mode, only the signals ending with “A” are used.**

## 8.2 Back I/O PMC Connector

Pin	Signal	Port	Pin	Signal	Port
1	CDA/-	0	33	CDA/-	2
2	CDB/+		34	CDB/+	
3	RXDA/-		35	RXDA/-	
4	RTSA/-		36	RTSA/-	
5	TXDA/-		37	TXDA/-	
6	CTSA/-		38	CTSA/-	
7	RTSB/+		39	RTSB/+	
8	CTSB/+		40	CTSB/+	
9	GND		41	GND	
10	TXDB/+		42	TXDB/+	
11	RXDB/+		43	RXDB/+	
12	TXCA/-		44	TXCA/-	
13	TXCB/+		45	TXCB/+	
14	GND		46	GND	
15	RXCA/-		47	RXCA/-	
16	RXCB/+		48	RXCB/+	
17	CDA/-	1	49	CDA/-	3
18	CDB/+		50	CDB/+	
19	RXDA/-		51	RXDA/-	
20	RTSA/-		52	RTSA/-	
21	TXDA/-		53	TXDA/-	
22	CTSA/-		54	CTSA/-	
23	RTSB/+		55	RTSB/+	
24	CTSB/+		56	CTSB/+	
25	GND		57	GND	
26	TXDB/+		58	TXDB/+	
27	RXDB/+		59	RXDB/+	
28	TXCA/-		60	TXCA/-	
29	TXCB/+		61	TXCB/+	
30	GND		62	GND	
31	RXCA/-		63	RXCA/-	
32	RXCB/+		64	RXCB/+	

Figure 8-3 : P14 I/O Pin Assignment

**In V.28 (single-ended) mode, only the signals ending with “A” are used.**