The Embedded I/O Company

TEWS TECHNOLOGIES

TMPE863

3 Channel High Speed
Sync/Async Serial Interface

Version 1.0

User Manual

Issue 1.0.0

January 2022

TEWS TECHNOLOGIES GmbH
Am Bahnhof 7 25469 Halstenbek, Germany
Phone: +49 (0) 4101 4058 0 Fax: +49 (0) 4101 4058 19
e-mail: info@tews.com www.tews.com

TMPE863-10R

3 Channel High Speed Sync/Async Serial Interface

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. RESET#.

Access terms are described as:

W	Write Only
R	Read Only
R/W	Read/Write
R/C	Read/Clear
R/S	Read/Set

Issue	Description	Date
1.0.0	Initial Issue	January 2022

Table of Contents

List of Figures

List of Tables

1 Product Description

The TMPE863 is a standard full PCI Express Mini Card, providing three high speed serial data communication channels.

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

Data transfer to and from host memory is handled via TMPE863 initiated DMA cycles for minimum host/CPU intervention.

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

Several serial communication protocols are supported for each channel, such as asynchronous (with oversampling), isochronous, synchronous and HDLC mode.

Available signal encodings for synchronous data communication are NRZ, NRZI, FM0, FM1 and Manchester.

On-Board clock generation provides a 14.7456 MHz clock for standard asynchronous baud rates, 10 MHz for the 10 Mbit/s synchronous data rate and 24 MHz for other baud or data rates.

Each channel provides various interrupt sources with support for MSI and legacy interrupts (INTA).

The Differential I/O lines for EIA-422, EIA-485 (Full-Duplex) are terminated with 120 Ω on-board. Receive Data (RxD +/-), Transmit Data (TxD +/-), Receive Clock (RxC +/-), Transmit Clock (TxC +/-) are available.

The I/O signals are accessible through a 30 pin Pico-Clasp latching connector.

The TMPE863 provides a basic heatsink to facilitate thermal management. The heatsink can be used to install additional cooling solutions like passive or active heatsinks or to provide a thermal connection to an enclosure.

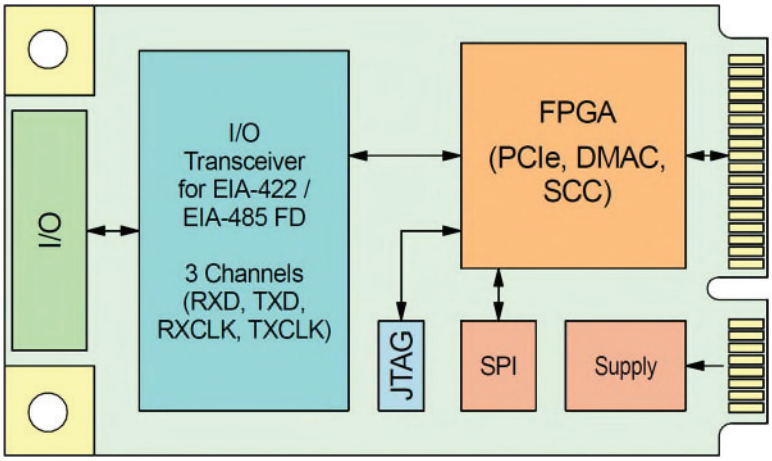


Figure 1-1 : Block Diagram

2 Technical Specification

PMC Interface		
Mechanical Interface	PCI Express Mini Card conforming to PCI Express Mini Card Electromech. Specification, Revision 2.0 Card Type: Full-Mini Card (50.95 mm x 30 mm)	
Electrical Interface	PCI Express x1 Link conforming to PCI Express Base Specification, Revision 2.1 (Xilinx) The TMPE863 does not support the USB interface	
Main On Board Devices		
FPGA	Xilinx Artix FPGA	
SPI Flash	128 Mbit SPI-Flash	
I/O Interface		
Number of Channels	3	
Signals per Channel	RXD±, TXD±, RXCLK±, TXCLK± (plus common GND)	
Digital I/O Interface	12 differential EIA-422 / EIA-485 I/O lines	
Digital I/O Transceiver	65HVD75D (or compatible)	
FIFOs	Main Transmit-FIFO per channel: up to 512 DWORD (2 Kbyte) Main Receive-FIFO per channel: 512 DWORD (2 Kbyte) SCC Transmit-FIFO per channel: up to 16 Byte SCC Receive-FIFO per channel: 16 Byte	
Maximum Data Rate	Synchronous: 10 Mbit/s (Non-DPLL Modes), 2 Mbit/s (DPLL Modes) Asynchronous: 2 Mbit/s	
I/O Connector	30 pos. Pico-Clasp latching connector	
Physical Data		
Power Requirements	+1.5V Supply: 200mA +3.3VAUX Supply: 425mA Idle, add appr. 90mA per active Channel (10MHz HDLC)	
Temperature Range	Operating Storage	-40°C to +85 °C -40°C to +125°C
MTBF	1.200.000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G_B 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.	
Humidity	5 – 95 % non-condensing	
Weight	13 g	

Table 2-1 : Technical Specification

3 DMA Controller

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

The interrupt handling / operation concept is based on Interrupt Queues located in host memory (interrupt vectors are transferred to cyclic interrupt queues in host memory).

Regarding the data FIFO structure, there is a common interrupt vector FIFO for all channels and channel dedicated FIFOs for transmit and receive direction.

The register map and register structure in the target register space is basically the same as on the TPMC863.

The data transferred by the DMA Controller is always processed/stored in little endian format.

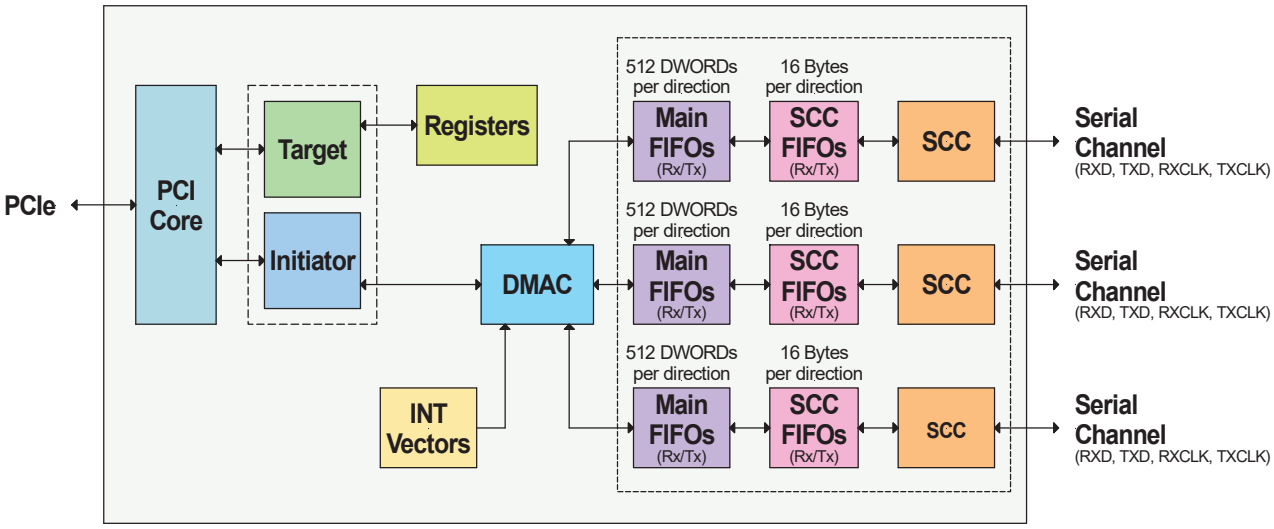


Figure 3-1 : FPGA On-Chip Block Diagram

The CPU prepares linked descriptor/buffer lists for transmit and receive channels in host 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 approach. The general strategy depends on the specific application. The DMA Controller (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. When the current descriptor has been processed, the DMAC is ready for branching to the next descriptor in the linked list. Because the DMA controller cannot distinguish between valid and invalid addresses, a control stall/wait mechanism is provided to prevent the DMA controller from branching to invalid memory locations after the current descriptor has been processed.

Two alternative control mechanisms are provided to handle descriptor list stall/wait conditions:

- Hold bit control mode

  After finishing a descriptor with active Hold bit, the DMAC waits until the Hold condition is released (before branching to the next descriptor).

- Last descriptor address control mode

  After finishing a descriptor whose address matches the address currently stored in the Last Descriptor Address Register, the DMAC waits until the address in the Last Descriptor Address Register no longer matches the current descriptor address (before branching to the next descriptor).

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

A 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 a HDLC frame.

The 'frame end' indications are stored in the internal FIFOs affecting 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 DMAC Transmit Descriptor Lists

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

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

Software writes the address of the first descriptor of each linked list to a dedicated Base Address Register (BTDAi) during the channel initialization procedure. Upon DMA transmitter initialization (IDT command), the corresponding DMA channel starts processing the descriptor list by fetching the first descriptor from this address. The DMAC temporarily stores the descriptor information (including the HOLD bit and Next Transmit Descriptor Address) as the current one and then moves the requested number of data bytes from the corresponding transmit data buffer to the channel's transmit FIFO. When this is done (depends on the amount of data and may also depend on the configured transmit FIFO size and transmit data rate) the DMAC sets the Complete bit in the fourth transmit descriptor word and is ready for branching to the next transmit descriptor in the linked list.

If the general DMA control mode is HOLD mode (CMODE='0') and the HOLD bit in the first descriptor word had been read as clear (HOLD='0') during the (initial) descriptor fetch phase, the DMAC branches to the next transmit descriptor (using the Next Transmit Descriptor Address that has been stored earlier in the descriptor fetch phase). If the HOLD bit in the first descriptor had been read as set (HOLD='1') during the (initial) descriptor fetch phase, the DMAC waits for a Transmit Poll Request (or Action Request with 'IDT' command) to be triggered by the user. When the Transmit Poll Request is triggered, the DMAC re-fetches the current descriptor and re-checks the HOLD bit found in the first descriptor word. If the HOLD bit is still set, the DMAC waits for another Transmit Poll Request. If the HOLD it is now clear, the DMAC branches to the most current Next Transmit Descriptor Address (i.e. the one that has been read during the descriptor re-fetch that has been triggered by the Transmit Poll Request).

If the general DMA control mode is LAST mode (CMODE='1'), the value in the LTDA register is being compared to the current descriptor address (CTDA) right before the next descriptor fetch is supposed to be started. If these addresses do not match, the DMAC branches to the next transmit descriptor (using the Next Transmit Descriptor Address that has been stored earlier in the descriptor fetch phase). If these addresses do match, the DMAC waits until the value in the LTDA register no longer matches the current descriptor address. When the value in the LTDA register no longer matches the current descriptor address, the current descriptor is re-fetched to re-obtain a most current Next Transmit Descriptor Address and then the DMAC branches to the next transmit descriptor.

3.1.1 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		

Table 3-1 : Transmit Descriptor (in Shared/Host Memory)

Transmit Descriptor DWORDs 0-2 are set by the user. Transmit Descriptor DWORD 3 is set by the DMAC (after descriptor completion).

FE: Frame End, set by the host

The FE bit indicates that the current transmit data section (addressed by Transmit Data Pointer) contains the end of a frame (HDLC) or the end of a 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.

In HDLC mode, the FE bit in the transmit descriptor controls the separation of the transmit data into HDLC frames.

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

The HOLD bit indicates whether the current descriptor is the last element of a linked list or not:

HOLD='0':

A next descriptor is available in host memory (and the Next Transmit Descriptor Pointer is assumed to be valid). After processing the current transmit descriptor, the DMAC branches to the next transmit descriptor without further ado.

HOLD='1':

The current descriptor is marked as the last one available for the DMAC. After processing the current transmit descriptor, the corresponding DMAC channel is deactivated for transmit direction as long as the host CPU does not request the DMAC to re-check the HOLD bit in the current transmit descriptor (in host memory) by the corresponding Transmit Poll Request bit in the GCMDR register (alternatively, the DMA transmitter may be reset and initialized again via the CHiCFG register).

HI: Host Initiated Interrupt

If the HI bit is set, the corresponding DMAC generates an interrupt with set HI bit when completing the transmit descriptor (i.e. when all data belonging to the transmit descriptor's data buffer has been fetched and copied to the main transmit FIFO).

NO: Number Of Bytes

This byte number defines the number of bytes stored in the data section to be transmitted. The maximum length of a 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.

Next Transmit Descriptor Pointer:

This 32-bit pointer contains the start address of the next transmit descriptor. After fetching the indicated number of data bytes of the current transmit descriptor, the DMAC is ready for branching to the next transmit descriptor to continue transmit data fetching. A transmit descriptor is read entirely and stored in on-chip memory. Therefore when the DMAC branches to a (next) descriptor all descriptor information must be valid. 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, the BTDA value in the BTDA register is used as a pointer for the first (next) transmit descriptor.

Transmit Data Pointer:

This 32-bit pointer contains the start address of the transmit data section for a transmit descriptor. Although the DMAC works long word oriented, it is possible to begin the transmit data section at byte addresses (note that receive data sections must always begin at a long word aligned address).

C: Complete

This bit is set by the DMAC if

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

3.2 DMAC Receive Descriptor Lists

Each receive descriptor consists of 5 consecutive DWORDs located DWORD aligned in host 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 address and 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 for descriptors containing the first 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 a 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 the initialization procedure. The corresponding DMA channel starts operating the linked lists at this address.

Software writes the address of the first descriptor of each linked list to a dedicated Base Address Register (BRDAi) during the channel initialization procedure. Upon DMA receiver initialization (IDR command), the corresponding DMA channel starts processing the descriptor list by fetching the first descriptor from this address. The DMAC temporarily stores the descriptor information as the current one and moves the received data bytes from the corresponding receive data FIFO to the receive descriptor's receive data buffer in host memory. The current receive descriptor/buffer is used until the buffer is full or a frame/block-end condition has been detected and the last data byte of the frame/block has been written to the receive buffer. When this is done the DMAC writes the appropriate information to the fourth transmit descriptor word and is then ready for branching to the next receive descriptor in the linked list.

If the general DMA control mode is HOLD mode (CMODE='0') and the HOLD bit in the current descriptor word had been read as clear (HOLD='0') during the (initial) descriptor fetch phase, the DMAC branches to the next receive descriptor (using the Next Receive Descriptor Address that has been stored during the descriptor fetch phase). If the HOLD bit in the first descriptor had been read as set (HOLD='1') during the (initial) descriptor fetch phase, then there is no way out of the HOLD condition other than a DMAC Receiver Reset via the RDR command bit (and Action Request).

If the general DMA control mode is LAST mode (CMODE='1'), the value in the LRDA register is being compared to the current descriptor address. If these addresses do not match, the DMAC branches to the next receive descriptor (using the Next Receive Descriptor Address that has been stored during the descriptor fetch phase). If these addresses do match, the DMAC waits until the value in the LRDA register no longer matches the current descriptor address. When the value in the LRDA register no longer matches the current descriptor address, the current descriptor is re-fetched to obtain the most current Next Receive Descriptor Address and then the DMAC branches to the next receive descriptor.

3.2.1 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					

Table 3-2 : Receive Descriptor (in Shared/Host Memory)

Receive Descriptor DWORDs 0-2 are set by the user. Receive Descriptor DWORDs 3 & (eventually) 4 are set by the DMAC (after descriptor completion).

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 host memory (and the Next Receive Descriptor Pointer is assumed to be valid). After processing the current receive descriptor, the DMAC branches to the next receive descriptor without further ado.

HOLD='1':

The current descriptor is marked as the last one available for the DMAC. After completion of the current receive descriptor, an ERR interrupt is generated (if not masked) and the corresponding DMAC channel is deactivated for receive direction. This state can only be released by an RDR (Reset DMA Receiver) command via the channel's CHiCFG register and the GCMDR register (for the Action Request).

HI: Host Initiated Interrupt

If the HI bit is set, the corresponding DMAC generates an interrupt with set HI bit when completing the receive descriptor (i.e. when the receive descriptor's data buffer has been filled completely or has been filled with a complete data frame (HDLC) or data block (ASYNC/ISYNC)).

NO: Number Of Bytes

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 the responsibility of the software. The maximum buffer length is 8188 bytes (i.e. NO = 0x1FFC).

Note that the receive data section may need to reserve space for up to 5 additional bytes in HDLC mode (32 bit CRC plus RSTA receiver status byte).

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 when the DMAC branches to a (next) descriptor all descriptor information must be valid.

Receive Data Pointer:

This 32-bit pointer contains the start address of the receive data section. The start address must be DWORD aligned.

Information written by the DMAC:

FE: Frame End

The FE bit indicates that the current receive data section (addressed by Receive Data Pointer) contains the end of a frame (HDLC) or the end of a data block (ASYNC). This bit is set by the DMAC after transferring the last data (of a data frame or data block) from the internal FIFO to the receive data section.

C: Complete

This bit is set by the DMAC if:

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

BNO: Number of Received Data Bytes

The DMAC writes the number of data bytes that were stored in the current data section into BNO (including CRC and status bytes).

STATUS: Receiver Status Byte

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.

Frame End Descriptor Pointer:

This 32-bit pointer is only valid in the descriptor that 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 data section (FE) of the HDLC frame or ASYNC block.

3.2.2 Receive Data Section Status Byte (HDLC Mode)

In HDLC protocol mode, the DMAC appends a Receive Status Byte (RSTA) to the receive data buffer (receive data buffer contains frame data bytes, CRC data bytes and the RSTA data byte). This status byte contains indications caused by the SCC (FIFO overflow, CRC result, Abort indication).

RSTA (HDLC):

7	6	5	4	3	2	1	0
1	**RFO**	**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 backward compatibility).

RFO: Receive FIFO Overflow

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

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.

3.2.3 Receive Data Section Status Byte (ASYNC Modes)

In ASYNC protocol mode a status byte can optionally be attached to every single data byte (CCR2.RFDF='1'), so that two bytes are stored for every received data byte.

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

15	14	13..9	8	7..0
parity error	frame error	reserved	parity bit	data byte

3.3 DMAC Interrupt Vector Queues

The interrupt concept is based on 32-bit interrupt vectors generated by the different blocks. Interrupt vectors are buffered in an on-board interrupt vector FIFO which is 32 DWORDs deep. The DMA controller transfers available interrupt vectors to one of seven circular interrupt queues located in the host memory, the actual queue depending on the source ID of the interrupt vector.

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

Each interrupt queue length and memory location can be configured via dedicated 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 interrupt queue (IQCFG).

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

Interrupt Queue	Assigned Interrupt Vectors
IQCFG	Configuration Interrupt Vectors
IQSCC0RX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 0 Receiver
IQSCC0TX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 0 Transmitter
IQSCC1RX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 1 Receiver
IQSCC1TX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 1 Transmitter
IQSCC2RX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 2 Receiver
IQSCC2TX	DMAC Interrupt Vectors & SCC Interrupt Vectors for Channel 2 Transmitter

Table 3-3 : Interrupt Queues and Vectors

3.3.1 Interrupt Vector Description

The following interrupt vector types are provided:

- Configuration Interrupt Vector
- DMAC Interrupt Vectors (per channel and direction)
- SCC Interrupt Vectors (per channel and direction)

3.3.1.1 Configuration Interrupt Vector

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

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

Table 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.

3.3.1.2 DMAC Interrupt Vectors

DMA controller interrupt vectors are transferred to the corresponding channel and direction specific interrupt queue (IQSCCiRX or IQSCCiTX respectively).

31	30..28	27:24	23..19	18	17	16	15..0
0	Source ID	00**0**0	0	HI	FI	ERR	0

Table 3-5 : DMA Interrupt Vector

Source-ID	Description
000	Receive Channel 0 Interrupt Vector (IQSCC0RX)
001	Receive Channel 1 Interrupt Vector (IQSCC1RX)
010	Receive Channel 2 Interrupt Vector (IQSCC2RX)
011	Reserved
100	Transmit Channel 0 Interrupt Vector (IQSCC0TX)
101	Transmit Channel 1 Interrupt Vector (IQSCC1TX)
110	Transmit Channel 2 Interrupt Vector (IQSCC2TX)
111	Reserved

Table 3-6 : DMAC Interrupt Vector Source-IDs

HI: Host Initiated interrupt:

This bit indicates that a Host Initiated (HI) interrupt occurred.

This interrupt can be activated by setting the 'HI' bit in the receive or transmit descriptor. In this case the DMAC will generate the HI-interrupt with completion of the current descriptor (when the DMAC is ready to branch to the next descriptor). This might be used to monitor the progress of the corresponding DMA channel on the descriptor linked 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.

FI: Frame Indication interrupt (Rx/Tx Channel)

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

This interrupt is generated with the completion of a receive or transmit descriptor that has the 'frame/block end' indication set (FE='1'), regardless if the DMAC branches to the next descriptor or remains in a HOLD/LAST condition.

For a transmit descriptor, it is generated when all data belonging to the transmit descriptor has been fetched from the transmit data buffer and copied to the transmit FIFO, and the FE bit is set in the transmit descriptor.

For a receive descriptor, it is generated when the receive data buffer of the receive descriptor has been filled with the last data belonging to a frame. The FE bit in the receive descriptor will also be set in this case.

A descriptor without a 'frame/block end' indication will generate an ERR (Error Indication) interrupt when reaching a HOLD/LAST condition after descriptor completion.

ERR: ERROR Indication interrupt (Rx/Tx Channel)

This bit indicates that an Error interrupt occurred.

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 (which are covered by SCC based interrupts RFO and XDU respectively).

Generated for the Receive direction if the Receive Descriptor runs into a HOLD/LAST condition after completion (and also when an ongoing Receiver DMAC operation is aborted by a DMAC Receiver Reset command).

Generated for the Transmit direction when the Transmit Descriptor runs into a HOLD/LAST condition while FE='0' or when the descriptor has a data size of NO=0 while FE='0'.

3.3.1.3 SCC Interrupt Vectors

Serial Channel (SCC) related interrupt vectors are transferred to the corresponding channel and direction specific interrupt queue (IQSCCiRX or 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 (ALLS, XDU) 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	RSV	RFO	0

Table 3-7 : SCC Interrupt Vector

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').

Source-ID	Description
000	Receive Channel 0 Interrupt Vector (IQSCC0RX)
001	Receive Channel 1 Interrupt Vector (IQSCC1RX)
010	Receive Channel 2 Interrupt Vector (IQSCC2RX)
011	Reserved
100	Transmit Channel 0 Interrupt Vector (IQSCC0TX)
101	Transmit Channel 1 Interrupt Vector (IQSCC1TX)
110	Transmit Channel 2 Interrupt Vector (IQSCC2TX)
111	Reserved

Table 3-8 : SCC Interrupt Vector Source-IDs

4 Serial Communication Controller

4.1 Protocol Modes

The following table provides an overview of the 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 actual protocol modes are selected via additional bit fields in registers CCR0 and CCR1.

	Register	CCR0	CCR1	CCR0
	Register Bit-Field	SM	MDS	BCR
Protocol Engine	**Protocol Mode**	Bit Settings		
HDLC	HDLC Address Mode 0	00	10	N/A
	Extended Transparent Mode		11	N/A
ASYNC	Asynchronous Mode	11	N/A	1
	Isochronous Mode		N/A	0

Table 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 Shared Flags and Zeros

The HDLC transmitter does not generate shared flags or shared zeroes between flags, while the HDLC receiver supports shared flags and shared zeroes between flags.

4.1.1.2 Address Mode 0

Standard HDLC framing and bit-stuffing is performed by the transmitter.
There is an option to append 16 bit or 32 bit CRC data for each frame.
For idle times, permanent flag transmission or constant high level is programmable.

The receiver will store the frame data (including CRC) plus a status byte.

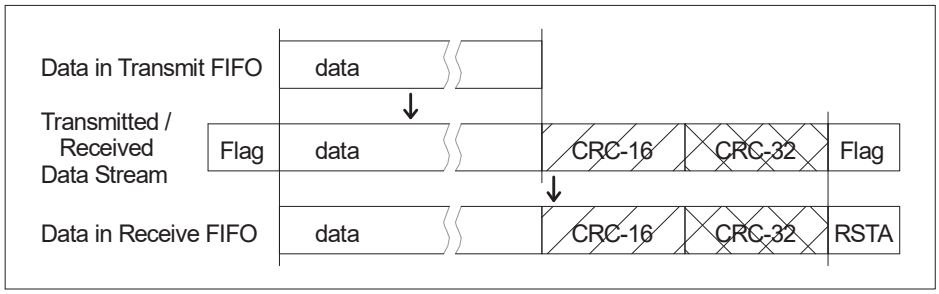


Figure 4-1 : HDLC Address Mode 0

4.1.1.3 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. Once enabled, the receiver gathers and stores the receive line state for every receive clock cycle.

4.1.2 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

NRZ data encoding and Bit clock rate x16 (register CCR0, bit BCR = ‘1’) shall be selected (register CCR0, bit field ’SC’).

The transmitter internally operates at a clock rate which is 16 times higher than the nominal data bit rate. The generated data bit rate is 1/16 of the transmitter’s clock rate.

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 x1 shall be selected (register CCR0 bit BCR = '0').

The isochronous mode uses the asynchronous character format including start/stop/parity bits. However, transmit data bits are generated at the transmitter clock rate and receive data bits are only sampled once (no oversampling).

Typically the transmit port provides transmit data and transmit clock signals to be used by the receive port. The input clock has to be phase locked to the data stream. This mode allows much higher transfer rates.

4.1.2.3 Data Transmission

Transmit data fetch from host memory starts after DMAC transmitter initialization. The data is first stored in the main transmit FIFO and is then forwarded to the dedicated SCC transmit FIFO.

Available character data bytes are taken from the SCC transmit FIFO and the character frame (start bit, character bits, optional parity bit and stop bits) is assembled. The character frame is transmitted on the TXD line (Character LSb first).

In isochronous mode, a transmit data bit (or line idle bit) is generated for every transmitter clock cycle. Both transmitter and receiver are operating with a clock signal frequency of the desired data rate.

In asynchronous mode the transmitter operates with a transmit clock of 16 times higher than the desired data rate (just as the receiver), however the transmit data bits are generated with an internally divided (by 16) transmit clock.

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

4.1.2.4 Data Reception

If the receiver is enabled, received data is stored in the channel's SCC receive FIFO (LSb first). Character length, number of stop bits and the optional parity bit are checked. Errors are indicated via interrupts. A character specific error status (framing and parity) can optionally be "attached" to each character in the receive FIFO.

There are two ways for terminating an asynchronous or isochronous block of data (that should not be mixed with other data blocks in the same receive descriptor/buffer):

- Using the Time-Out function to detect a programmable amount of idle level time after a character reception (Bit 'TOIE' and bitfield 'TOLEN' in the channel's CCR2 register). This requires an adequate pause/idle time between transmissions by the transmitter.
- Using the programmable Termination Character function (bit 'TCDE' and bit field 'TC' in the channel's TCR register)

There are two ways for controlling the forwarding of gathered receive data to the receive descriptors data buffer in host memory:

- Receive FIFO Threshold (see bit field 'RFTH' in the channel's CCR2 register)
- Receive FIFO Data Format (see bit 'RFDF' in the channel's CCR2 register)

4.1.2.5 Break Detection/Generation

Break generation (Transmitter):

Upon issuing the transmit break command (bit ’XBRK’ in register CCR2), the TxD pin is forced to physical ‘0’ level with the next following transmit clock edge, and is released again with the first transmit clock edge after this condition is released again by software.

Break detection (Receiver):

The SCC receiver recognizes the break condition upon receiving consecutive (physical) ‘0’s for the defined character length, optional parity bit and selected number of stop bits. Such ‘zero’ characters are discarded and are not pushed to the Receive FIFO. The ’Break’ interrupt status (BRK) is generated, if enabled. The break condition will be present until a ‘1’ is received which is then indicated by the ‘Break Termination’ interrupt status (BRKT).

4.2 Clock Sources

The TMPE863 supports multiple clock sources for the transmitter and receiver circuits, controlled by the ACR register. Clock source options are three base clock frequencies (14.7456 MHz, 24 MHz and 10 MHz), the external RxCLK and TxCLK inputs or the internal clock recovery circuit (DPLL). TxCLK can be an input for the internal transmit clock or an output providing a transmit clock signal (see figure).

The maximum data rate is 10Mbit/s for synchronous mode when the DPLL is not used. The maximum data rate in asynchronous mode with oversampling or synchronous DPLL mode is 2 Mbit/s (with the BRG output clock at 16 times higher than the data rate for both). To generate higher internal clock frequencies for oversampling or DPLL reference clock, an optional x4 clock multiplier is provided.

The input frequency range of the x4 clock multiplier is 10 MHz to 24 MHz.
These values must never be exceeded to ensure proper function of the clock multiplier.

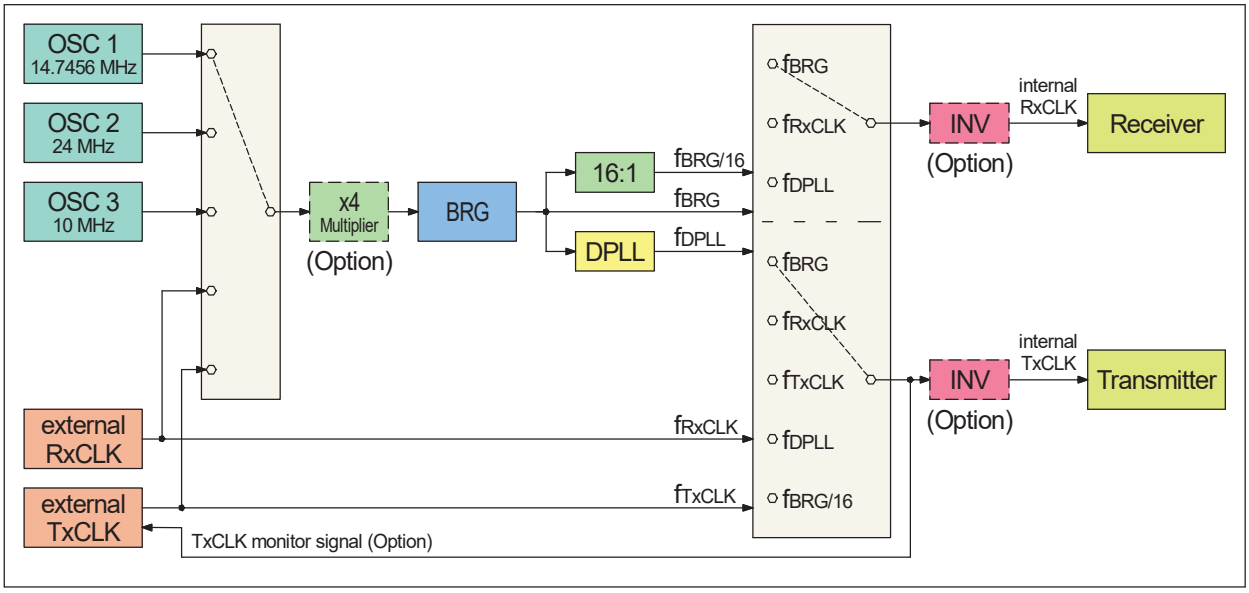


Figure 4-2 : Clock Sources

4.3 Baud Rate Generation

There is a dedicated Baud Rate Generator (BRG) for each of the three channels (BRG registers 0x012C, 0x01AC, 0x022C).

The Baud Rate Generator output clock frequency is: $f_{BRG} = f_{in} / k$.

The baud rate generator input clock f_{in} depends on the selected clock source (see also previous chapter “Clock Sources”).

The divisor k can be set in 2 ways, determined by BRR[31].

When BRR[31] = 0, k is calculated as:

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

with N (BRR[5:0]) = 0..63 and M (BRR[11:8]) = 0..15

The alternative is to set (k – 1) directly as a 21-bit wide value, when BRR[31] = 1:

$$k = BRR[20:0] + 1$$

For standard Asynchronous serial mode as well as for using the DPLL as a clock source, the baud rate generator output clock shall be set to 16 times the expected/desired data rate.

4.4 Data Encoding

The following encodings of the serial data are supported per channel:

- 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)

NRZ data encoding must be used for Asynchronous mode.

FM0, FM1 and Manchester encoding typically require the DPLL as the receiver clock source.

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.

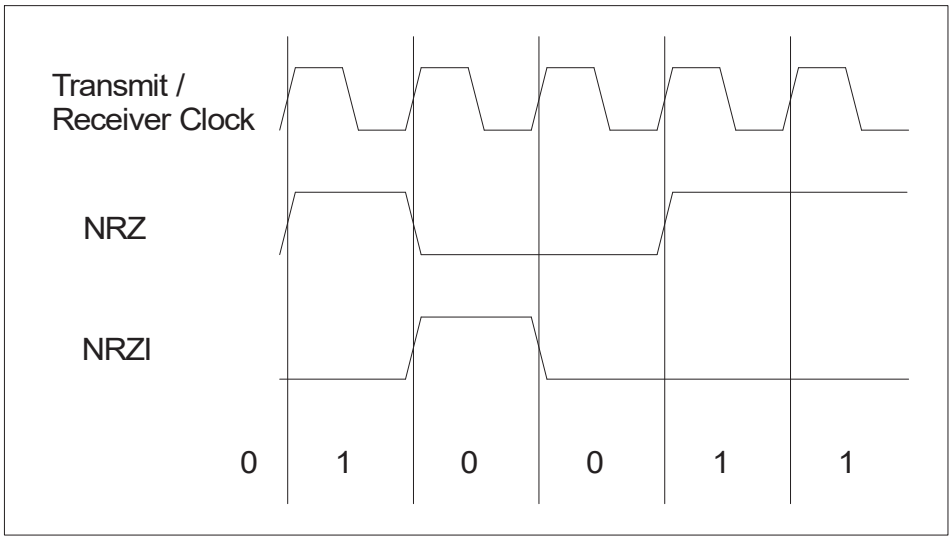


Figure 4-3 : NRZ and NRZI Data Encoding

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, whereas a logical ‘0’ has none. The transmit clock precedes the receive clock by 90°.

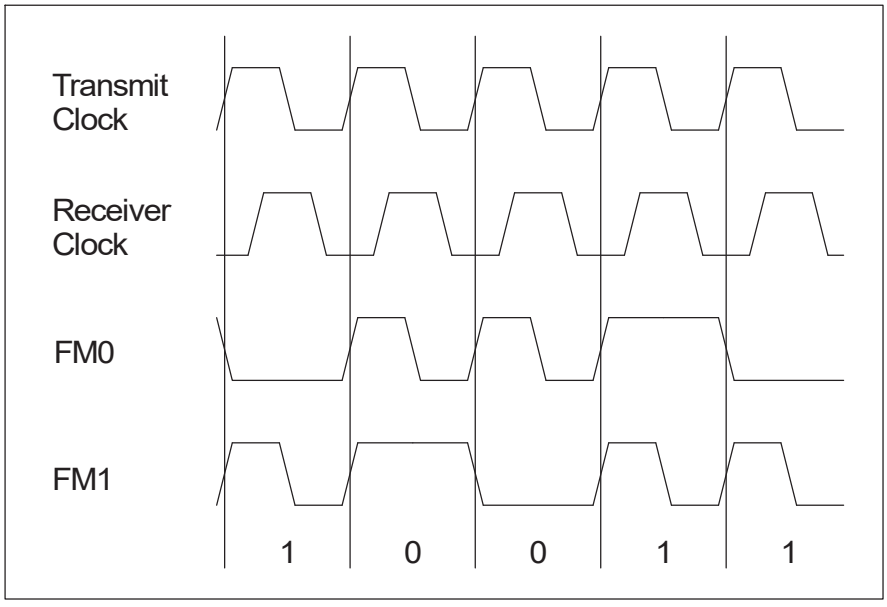


Figure 4-4 : FM0 and FM1 Data Encoding

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°.

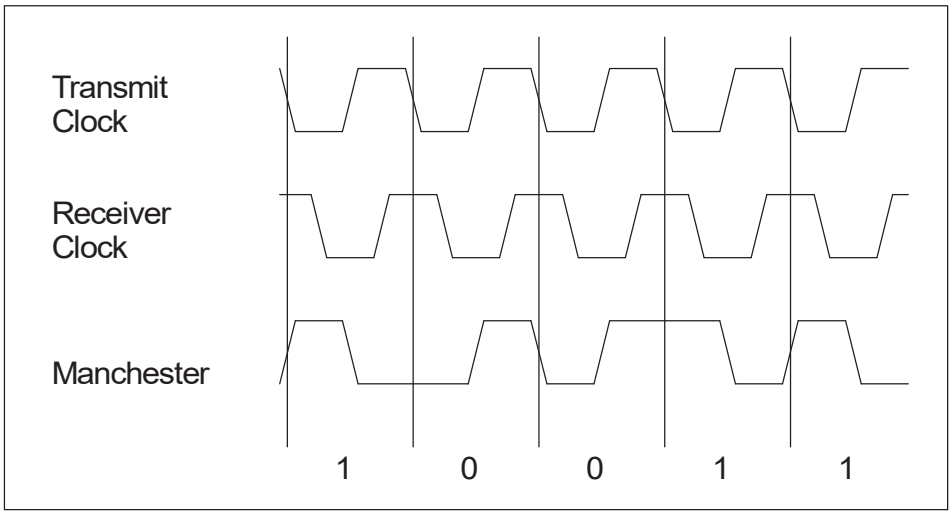


Figure 4-5 : Manchester Data Encoding (on TMPE863)

4.5 Clock Recovery (DPLL)

FM0, FM1 and Manchester data encodings are eliminating the need to transfer additional clock information via a separate serial clock line (so there is no additional clock signal transferred along with the data stream). Instead the clock signal is recovered from the receive data stream by the use of the internal DPLL circuit.

The main task of the DPLL (digital-phase-locked-loop) is to derive the receive clock from the incoming data stream and to adjust its phase to the incoming data in order to enable optimal bit sampling.

The DPLL reference clock is the baud rate generator output clock which must be set to be 16 times higher than the expected data rate. The receive clock source must be set to ‘DPLL’ (ACR.RCS = ‘10’). The transmit clock source may be set to be the baud rate generator output clock divided by 16 (ACR.TCS = ‘100’) or to be the transmit clock generated by the DPLL circuit (ACR.TCS = ‘011’).

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

5 Address Map

5.1 PCI Identifier

Vendor ID	0x1498 (TEWS Technologies)
Device ID	0xA35F (TMPE863)
Class Code	0x028000 (Other Network Controller)
Subsystem Vendor ID	0x1498 (TEWS Technologies)
Subsystem ID	0xA00A (TMPE, -10R)

Table 5-1 : PCI Identifier

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

Table 5-2 : PCI Base Address Registers

6 Register Description

6.1 PCI Memory Space Registers Overview

Offset to PCI 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…0x07FF	0	Reserved

Table 6-1 : Local Register Space

6.2 Global Registers Overview

All registers are 32 bit organized.

Offset to PCI Base Address BAR0	Register Name	
0x0000	GCMDR	Global Command Register
0x0004	GSTAR	Global Status Register
0x0008	GMODE	Global Mode Register
Interrupt Queue IQ specific registers (+ FIFO Control 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	Reserved	Reserved
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	Reserved	Reserved
0x0034	FIFOCR4	FIFO Control Register 4
0x0038	Reserved	-
0x003C	IQCFGBAR	IQ CFG Base Address Register
0x0040	Reserved	-
0x0044	FIFOCR1	FIFO Control Register 1
0x0048	Reserved	-
0x004C	Reserved	-

Offset to PCI Base Address BAR0	Register Name	
DMA Controller (DMAC) specific registers		
0x0050	CH0CFG	Channel 0 Configuration Register
0x0054	CH0BRDA	Channel 0 First/Base Rx Descr. Address
0x0058	CH0BTDA	Channel 0 First/Base Tx Descr. Address
0x005C	CH1CFG	Channel 1 Configuration Register
0x0060	CH1BRDA	Channel 1 First/Base Rx Descr. Address
0x0064	CH1BTDA	Channel 1 First/Base Tx Descr. Address
0x0068	CH2CFG	Channel 2 Configuration Register
0x006C	CH2BRDA	Channel 2 First/Base Rx Descr. Address
0x0070	CH2BTDA	Channel 2 First/Base Tx Descr. Address
0x0074	Reserved	Reserved
0x0078	Reserved	Reserved
0x007C	Reserved	Reserved
0x0080...0x0097	Reserved	-
0x0098	CH0CRDA	Channel 0 Current Rx Descr. Address
0x009C	CH1CRDA	Channel 1 Current Rx Descr. Address
0x00A0	CH2CRDA	Channel 2 Current Rx Descr. Address
0x00A4	Reserved	Reserved
0x00A8	Reserved	-
0x00AC	Reserved	-
0x00B0	CH0CTDA	Channel 0 Current Tx Descr. Address
0x00B4	CH1CTDA	Channel 1 Current Tx Descr. Address
0x00B8	CH2CTDA	Channel 2 Current Tx Descr. Address
0x00BC	Reserved	Reserved
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	Reserved	Reserved
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	Reserved	Reserved
Other registers		
0x00F0	VR	Version Register

Offset to PCI Base Address BAR0	Register Name	
0x00F4	ISPR	ISP Register (Factory Reserved)
0x00F8	GCTLR	Global Control Register
0x00FC...0x00FF	Reserved	-

Table 6-2 : Global Registers

6.3 SCC Registers Overview

All registers are 32-bit organized.

The SCC registers are used to configure and control the three Serial Communication Controllers (SCCs). There is a complete SCC register set for every SCC channel.

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

• Register address offset within the SCC register set (range 0x00 ...0x5C)

Most registers and register bit positions are shared by both main SCC protocol modes (HDLC, ASYNC). However the meaning (and name) of individual bit positions might differ for the different protocol modes.

Offset to Base Address 0x0100 (SCC0) 0x0180 (SCC1) 0x0200 (SCC2)	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

Offset to Base Address 0x0100 (SCC0) 0x0180 (SCC1) 0x0200 (SCC2)	Register Name	
0x5C	ACR	Additional Configuration Register
0x60...0x7F	reserved	-

Table 6-3 : SCC Registers

6.4 Global Registers

6.4.1 GCMDR – Global Command Register (0x0000)

Bit	Symbol	Description	Access	Reset Value
31	-	Reserved	R/W	0
30	CFGIQSCC2RX	Configure SCC2 Receive Interrupt Queue (see below)	R/W	0
29	CFGIQSCC1RX	Configure SCC1 Receive Interrupt Queue (see below)	R/W	0
28	CFGIQSCC0RX	Configure SCC0 Receive Interrupt Queue Only evaluated if action request bit 'AR' is set. The DMA (interrupt) controller will transfer interrupt vectors generated by the dedicated SCC receiver (0 to 2) to the corresponding interrupt queue which must be configured via the 'CFGIQSCCiRX' command bits: '0': The DMA (interrupt) controller does NOT configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'. '1': Causes the DMA (interrupt) controller to configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'. Upon 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. This bit will be cleared automatically upon a successful action request.	R/W	0
27	-	Reserved	R/W	0
26	CFGIQSCC2TX	Configure SCC2 Transmit Interrupt Queue (see below)	R/W	0
25	CFGIQSCC1TX	Configure SCC1 Transmit Interrupt Queue (see below)	R/W	0

Bit	Symbol	Description	Access	Reset Value
24	CFGIQSCC0TX	Configure SCC0 Transmit Interrupt Queue Only evaluated if action request bit 'AR' is set. The DMA (interrupt) controller will transfer interrupt vectors generated by the dedicated SCC transmitter (2..0) to the corresponding interrupt queue which must be configured via the 'CFGIQSCCiTX' command bits: '0': The DMA (interrupt) controller does NOT configure/re-configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'. '1': Causes the DMA (interrupt) controller to configure the corresponding interrupt queue, if action request bit 'AR' is set to '1'. Upon 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. This bit will be cleared automatically upon a successful action request.	R/W	0
23:22	-	Reserved (0 for reads)	R	0
21	CFGIQCFG	Configure Configuration Interrupt Queue Only evaluated if action request bit 'AR' is set. The DMA (interrupt) controller will transfer action request acknowledge/failure interrupt vectors to the configuration interrupt queue which must be configured via the 'CFGIQCFG' command bit: '0': The DMA (interrupt) controller does NOT configure/re-configure the configuration interrupt queue, if action request bit 'AR' is set to '1'. '1': Causes the DMA (interrupt) controller to configure the configuration interrupt queue, if action request bit 'AR' is set to '1'. Upon 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. This bit will be cleared automatically upon a successful action request.	R/W	0
20:14	-	Reserved (0 for reads)	R	0
13	-	Reserved	R/W	0
12	TXPR2	Transmit Poll Request Channel 2	R/W	0
11	TXPR1	Transmit Poll Request Channel 1	R/W	0

Bit	Symbol	Description	Access	Reset Value
10	TXPR0	Transmit Poll Request Channel 0 Self-clearing command bit, only valid in 'HOLD' bit controlled DMA controller mode (bit CMODE = '0' in register GMODE): '0': No Transmit Poll Request is performed. The corresponding DMA controller transmit channel is stopped after descriptor completion when HOLD='1' has been detected in the current transmit descriptor. '1': Setting this bit to '1', when HOLD='1' has been detected in the current transmit descriptor, will cause the controller to (re-)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 transmit descriptor again. If the 'HOLD' bit is detected cleared ('0'), the DMA controller will branch to the next descriptor. When the channel's DMA controller is not in 'HOLD' state, this command is discarded.	R/W	0
9	IMAR	Interrupt Mask Action Request	R/W	1
8:1	-	Reserved (0 for reads)	R	0
0	AR	Action Request Self-clearing command bit. '0': No action request '1': Action request Validates GCMDR CFGIQSCCiRX, CFGIQSCCiTX, CFGIQCFG register bits and CHiCFG RDR, RDT, IDR, IDT register bits.	R/W	0

Table 6-4 : Global Command Register

6.4.2 GSTAR – Global Status Register (0x0004)

Status set by DMA Controller as an 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 the interrupt (legacy PCI INTA interrupt) is active.

Seven interrupt queues are provided:

– three receive interrupt vector queues (one for each SCCi, i=0...2)

– three transmit interrupt vector queues (one for each SCCi, i=0...2)

– one configuration interrupt vector queue (action request acknowledge/fail)

To clear a bit in the status register, the host CPU must write a '1' to the corresponding register bit. The interrupt (legacy PCI INTA interrupt) will be asserted if any of the GSTAR interrupt indication bits is set. The interrupt (legacy PCI INTA interrupt) signal INTA# will be de-asserted if all GSTAR interrupt indication bits are cleared.

Bit	Symbol	Description	Access	Reset Value
31	-	Reserved	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	-	Reserved	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
Action Request Result Status				
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

Table 6-5 : Global Status Register

See chapter 'Interrupt Vector Description' for a description of the interrupt vector structure.

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 (in host memory) '1' Last Receive/Transmit Descriptor Address mode. The descriptor chain end condition is controlled via the LRDA/LTDA registers.	R/W	0

Table 6-6 : Global Mode Register

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

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

Table 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	(Cyclic) Configuration Interrupt Queue Length Queue Length = (1 + 'IQCFGLEN') * 32 DWORDS, 'IQCFGLEN' = 0...15	R/W	0
19:0	-	Reserved (0 for reads)	R	0

Table 6-8 : Interrupt Queue Length Register 1

6.4.6 IQSCCiRXBAR – Interrupt Queue SCCi Receiver Base Address Register (i=0…2) (0x0014, 0x0018, 0x001C)

Bit	Symbol	Description	Access	Reset Value
31:2	IQSCCiRXBAR	PCI Base Address of the SCC channel's Receive Interrupt Queue	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-9 : IQSCCiRXBAR Register

6.4.7 IQSCCiTXBAR – Interrupt Queue SCCi Transmitter Base Address Register (i=0...2) (0x0024, 0x0028, 0x002C)

Bit	Symbol	Description	Access	Reset Value
31:2	IQSCCiTXBAR	PCI Base Address of the SCC channel's Transmit Interrupt Queue	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-10 : IQSCCiTXBAR Register

6.4.8 FIFOCR4 – FIFO Control Register 4 (0x0034)

Bit	Symbol	Description	Access	Reset Value
31:24	Reserved	Reserved	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
7:0	TFFTHRES0	Transmit FIFO Forward Threshold Channel 0 These bit fields determine the channel specific Transmit FIFO Forward Threshold for the corresponding channel in number of DWORDs. This threshold controls DMAC operation towards the serial communication controllers (SCC). As soon as the number of valid data words (belonging to the same frame) in the main transmit FIFO is greater than the threshold, the DMAC will provide transmit data to the corresponding SCC. Once having started serving data for a frame, the DMAC will ignore this threshold providing all available data of the current frame to the SCC. Threshold operation starts again with the next frame. Frames shorter than the threshold will be transferred as soon as a frame end indication is detected by the DMAC. *Note: Intended for frame oriented data transmission (i.e. HDLC). 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.*	R/W	0

Table 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	PCI Base Address of the Configuration Interrupt Queue	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-12 : IQCFGBAR Register

6.4.10 FIFOCR1 – FIFO Control Register 1 (0x0044)

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>31:27</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>26:24</td><td>Reserved</td><td>Reserved</td><td>R/W</td><td>111</td></tr>
<tr><td>23:19</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>18:16</td><td>TFSIZE2</td><td>Main Transmit-FIFO Size (Depth) Channel 2</td><td>R/W</td><td>111</td></tr>
<tr><td>15:9</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>10:8</td><td>TFSIZE1</td><td>Main Transmit-FIFO Size (Depth) Channel 1</td><td>R/W</td><td>111</td></tr>
<tr><td>7:3</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>2:0</td><td>TFSIZE0</td><td>Main Transmit-FIFO Size (Depth) Channel 0
Main Transmit-FIFO Size (Depth) is:
2 power (TFSIZEi + 2)
<table><tr><th>TFSIZEi</th><th>FIFO Size (Depth) in DWords (4 Bytes)</th></tr><tr><td>111</td><td>512</td></tr><tr><td>110</td><td>256</td></tr><tr><td>101</td><td>128</td></tr><tr><td>100</td><td>64</td></tr><tr><td>011</td><td>32</td></tr><tr><td>010</td><td>16</td></tr><tr><td>001</td><td>8</td></tr><tr><td>000</td><td>4</td></tr></table>Note: The effective overall Transmit-FIFO size is slightly larger than the Main Transmit-FIFO size because of an additional small SCC FIFO (up to 16 Bytes).</td><td>R/W</td><td>111</td></tr>
</table>

Table 6-13 : FIFO Control Register 1

6.4.11 CHiCFG – Channel i Configuration Register (i=0...2) (0x0050, 0x005C, 0x0068)

Bit	Symbol	Description	Access	Reset Value
31:28	-	Reserved (0 for reads)	R	0
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=0...2): '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=0...2): '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=0...2): '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=0...2): '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

Bit	Symbol	Description	Access	Reset Value
22	RDR	Reset DMA Receiver (Channel i) Must be validated by the Action Request bit in the GCMDR. Cleared automatically if successful. 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
21	RDT	Reset DMA Transmitter (Channel i) Must be validated by the Action Request bit in the GCMDR. Cleared automatically if successful. 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) Must be validated by the Action Request bit in the GCMDR. Cleared automatically if successful. This command causes the specific DMA receive channel to load the first/base descriptor address from register CHiBRDA and to fetch the corresponding descriptor. Afterwards normal DMA operation on the receive descriptor list is performed depending on the selected DMA control mode. *Note: To avoid unexpected DMA controller behavior, it is recommended to apply 'IDR' command only, if the specific DMA channel is in reset state.*	R/W	0
19	IDT	Initialize DMA Transmitter (Channel i) Must be validated by the Action Request bit in the GCMDR. Cleared automatically if successful. This command causes the specific DMA transmit channel to load the first/base descriptor address from register CHiBTDA and to fetch the corresponding descriptor. Afterwards normal DMA operation on the transmit descriptor list is performed depending on the selected DMA control mode. *Note: To avoid unexpected DMA controller behavior, it is recommended to apply 'IDT' command only, if the specific DMA channel is in reset state.*	R/W	0
18:0	-	Reserved (0 for reads)	R	0

Table 6-14 : CHiCFG Register

See chapter 'Interrupt Vector Description' for a description of the interrupt vector structure.

6.4.12 CHiBRDA – Channel i First/Base Receive Descriptor Address Register (i=0...2) (0x0054, 0x0060, 0x006C)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiFRDA	PCI Base Address of Receive Descriptor	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-15 : CHiBRDA Register

6.4.13 CHiBTDA – Channel i First/Base Transmit Descriptor Address Register (i=0...2) (0x0058, 0x0064, 0x0070)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiFTDA	PCI Base Address of Transmit Descriptor	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-16 : CHiBTDA Register

6.4.14 CHiCRDA – Channel i Current Receive Descriptor Address Register (i=0...2) (0x0098, 0x009C, 0x00A0)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiCRDA	DMAC enters the PCI Base Address of the current Receive Descriptor	R	0
1:0		Reserved (0 for reads)	R	0

Table 6-17 : CHiFRDA Register

6.4.15 CHiCTDA – Channel i Current Transmit Descriptor Address Register (i=0...2) (0x00B0, 0x00B4, 0x00B8)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiCTDA	DMAC enters the PCI Base Address of the current Transmit Descriptor	R	0
1:0		Reserved (0 for reads)	R	0

Table 6-18 : CHiFTDA Register

6.4.16 CHiLRDA – Channel i Last Receive Descriptor Address Register (i=0...2) (0x00C8, 0x00CC, 0x00D0)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiLRDA	PCI Base Address of Last Receive Descriptor Only applicable when the DMA Control Mode is set to Last Descriptor Address Mode in the GMODE register.	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-19 : CHiLRDA Register

6.4.17 CHiLTDA – Channel i Last Transmit Descriptor Address Register (i=0...2) (0x00E0, 0x00E4, 0x00E8)

Bit	Symbol	Description	Access	Reset Value
31:2	CHiLTDA	PCI Base Address of Last Transmit Descriptor Only applicable when the DMA Control Mode is set to Last Descriptor Address Mode in the GMODE register.	R/W	0
1:0		Reserved (0 for reads)	R	0

Table 6-20 : CHiLTDA Register

6.4.18 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	FPGA Logic Version	R	

Table 6-21 : Version Register

6.4.1 ISPR – In-System-Programming Register (0x00F4)

Bit	Symbol	Description	Access	Reset Value
31:8	-	Reserved (0 for reads)	R	0
7	ISP_EN	Enables the SPI_DI and SPI_SEL outputs when set.	R/W	0
6:4	-	Reserved (0 for reads)	R	000
3	SPI_DO	Indicates the state of the SPI Flash Data Output line	R	0
2	SPI_DI	Controls the state of the SPI Flash Data Input line	R/W	0

1	SPI_SEL	Controls the state of the SPI Flash Select line	R/W	0
0	SPI_CLK	Controls the state of the SPI Flash Clock line	R/W	0

Table 6-22 : ISPR Register

This register is reserved for factory (re-) programming of the FPGA configuration Flash Memory.

No user write access shall be done to this address, as a corrupted FPGA configuration Flash Memory would prevent successful FPGA configuration at power-up.

6.4.2 GCTLR – Global Control Register (0x00F8)

Bit	Symbol	Description	Access	Reset Value
31:20	-	Reserved (0 for reads)	R	0
19	EE_SDA(i)	Serial I2C EEPROM Data Line (Input)	R	-
18	EE_SDA(o)	Serial I2C EEPROM Data Line (Output) 0: I2C Data Signal is actively driven low (requires that EE_CS Bit is set to '1') 1: I2C Data Signal is tri-stated (high level set by pull-up resistor)	R/W	0
17	EE_CS	Serial I2C EEPROM Select and SDA Output Enable	R/W	0
16	EE_SCL	Serial I2C EEPROM Clock 0: I2C Clock Signal is actively driven low 1: I2C Clock Signal is tri-stated (high level set by pull-up resistor)	R/W	0
15:8	RETRYCNT	Maximum number of retries, that occurred during a PCI transaction Not supported on TMPE863.	R	0x00
7	RCNTCLR	Reset maximum number of retries Self-clearing command bit Not supported on TMPE863.	R/W	0
6	INI_HALT	Initiator State Machine is stopped due to PCI transaction abort	R	0
5	INI_REL	Release the stopped Initiator State Machine Self-clearing command bit	R/W	0
4:1	-	Reserved	R	0
0	LRST	Local Reset Self-clearing command bit. The complete local part of the device is reset (application registers included). Only the registers in the PCI configuration space keep their values.	R/W	0

Table 6-23 : Global Control Register

6.5 SCC Channel Specific Registers

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

Bit	Symbol	Description	Access	Reset Value
31:25	-	Reserved (0 for reads)	R	0
24	XRES	Transmitter Reset (Self-clearing) '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). *Note: A transmit clock must be present.*	R/W	0
23:17	-	Reserved (0 for reads)	R	0
16	RRES	Receiver Reset (Self-clearing) '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). *Note: A receive clock must be present.*	R/W	0
15:0	-	Reserved (0 for reads)	R	0

Table 6-24 : Command Register

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

Bit	Symbol	Description	Access	Reset Value
31:25	-	Reserved (0 for reads)	R	0
24	CTS	Not supported on TMPE863	R	-
23:22		Reserved (0 for reads)	R	0
21	CD	Not supported on TMPE863	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	Not supported on TMPE863	R	-

Table 6-25 : Status Register

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

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 *Note: Asynchronous (with oversampling) protocol requires NRZ data encoding. FM0, FM1 and Manchester encoding typically requires using the DPLL as receiver clock source and a baud rate generator output clock of approx. 16x data bit rate.* *Note: Using FM1 and Manchester Encoding may require a dummy data transfer for correct synchronization.*	R/W	000
19:18	-	Reserved (0 for reads)	R	0
17:16	SM	Serial Port Mode Selects the protocol engine: '00': HDLC synchronous '01': reserved '10': reserved '11': Asynchronous / Isochronous	R/W	00
15:13	-	Reserved (0 for reads)	R	0

Bit	Symbol	Description	Access	Reset Value
12	VIS	Masked Interrupts Visible '0': Masked interrupt status bits are not visible on interrupt status register (ISR) read access. '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. *Note: Masked interrupts will not generate an interrupt vector to the interrupt controller.*	R/W	0
11:8	-	Reserved (0 for reads)	R	0
7	BCR	Bit Clock Rate (async/isochr) '0': Isochronous (Bit Clock Rate x1). I.e. Asynchronous without oversampling. Transmitter/Receiver Clock Rate is Data Bit Rate x1. Bits are sampled once. '1': Standard Asynchronous (Bit Clock Rate x16). I.e. Asynchronous with 16x oversampling. Receiver Clock Rate is Expected Data Bit Rate x16. Bits are sampled 16 times. The result is determined by a majority decision of 3 samples around the bit center. Transmitter Clock Rate is Data Bit Rate x16. Effective Transmit Data Bit Rate is Transmitter Clock Rate /16. 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 an input (DCE mode) '1': TxC is an output (DTE# mode) *Note: The Transceiver direction of TxC is set according to TOE.*	R/W	0
4:0	-	Reserved (0 for reads)	R	0

Table 6-26 : Channel Configuration Register 0

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

Bit	Symbol	Description	Access	Reset Value
31:21	-	Reserved (0 for reads)	R	0
20	RTS	Not supported on TMPE863	R/W	0
19	FRTS	Not supported on TMPE863	R/W	0
18	FCTS	Not supported on TMPE863	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
8	TLP	Test Loop In Test Loop mode, the internal RXD signal is detached from the RXD I/O Pin and connected to the internal TXD signal instead (RXD = TXD). '0': Test loop disabled. '1': Test loop enabled. *Note: The Test Loop is only effective for the data line (not for the clock line). TXD is still driven out in Test Loop mode.*	R/W	0
7	TOIE	Time Out Indication Enable (async./isochr.) A 'block end' indication is inserted in the receive FIFO if a time out occurs. The current receive descriptor will be terminated. '0': Time Out function disabled. '1': Time Out function enabled. *Note: A time out event will generate a 'TIME' interrupt (if unmasked).*	R/W	0
6:0 (asyn)	TOLEN	Time Out Length (async./isochr.) 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'. The period of time is programmable in multiples of **a single** character frame length (CFL) time equivalents including start, parity and stop bits: T_{TOUT} = (TOLEN + 1) * CFL	R/W	0

Bit	Symbol	Description	Access	Reset Value
1 (hdlc)	CRL	CRC Reset Value (hdlc) Defines the initial value for the CRC generators: '0': Initial value is 0xFFFF (16 bit CRC), 0xFFFFFFFF (32 bit CRC); (default value for most HDLC applications) '1': Initial value is 0x0000 (16 bit CRC), 0x00000000 (32 bit CRC).	R/W	0
0 (hdlc)	C32	CRC-32 Select (hdlc) This bit selects 32-bit CRC operation for transmit and receive. '0': 16-bit CRC generation/checking. '1': 32-bit CRC generation/checking. CRC-16 Polynomial used is $x^{16} + x^{12} + x^5 + 1$. CRC-32 Polynomial used is $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	R/W	0

Table 6-27 : Channel Configuration Register 1

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

Bit	Symbol	Description	Access	Reset Value
31:30	-	Reserved (0 for reads)	R	0
29:28	CHL	Character Length (async./isochr.) '00': 8 bit data '01': 7 bit data '10': 6 bit data '11': 5 bit data *Note: Transmit and receive data is always passed byte oriented on the host interface. E.g. for 5 bit character length there are 3 don't care bits in each data byte read from or written to host memory buffers.*	R/W	00
27	RAC	Receiver active '0': Receiver inactive, receive line is ignored. '1': Receiver active.	R/W	0
26	-	Reserved (0 for reads)	R	0
25	XBRK	Transmit Break (async./isochr.) '0': Normal transmit operation '1': Forces the TxD pin to 'low' level immediately (break condition), regardless of any character being currently transmitted. This command is executed immediately with the next rising edge of the transmit clock and further transmission is disabled. The currently sent character is lost. Data stored in the SCC transmit FIFO will be sent as soon as the break condition is cleared (XBRK='0'). A transmit reset command (bit 'XRES' in register CMDR) does NOT clear the break condition automatically.	R/W	0
24 (asyn)	STOP	Stop Bit Number (async./isochr.) '0': 1 stop bit per character '1': 2 stop bits per character	R/W	0
23:22 (asyn)	PAR	Parity Format (async./isochr.) '00': Space ('0') is inserted as parity bit '01': Odd parity '10': Even parity '11': Mark ('1') is inserted as parity bit *Note: The received parity bit (and parity error) is stored in the receive data buffer if bit 'RFDF' = '1'.*	R/W	0
21 (asyn)	PARE	Parity Enable (async./isochr.) '0': Parity generation/checking is disabled. '1': Parity generation/checking is enabled.	R/W	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Description</th><th>Access</th><th>Reset Value</th></tr>
<tr><td>24:23
(hdlc)</td><td>-</td><td>Reserved (0 for reads) (hdlc)</td><td>R</td><td>0</td></tr>
<tr><td>22
(hdlc)</td><td>DRCRC</td><td>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.
Note: A received checksum (2 or 4 bytes) is always forwarded to the receive buffer as data.</td><td>R/W</td><td>0</td></tr>
<tr><td>21
(hdlc)</td><td>-</td><td>Reserved (0 for reads) (hdlc)</td><td>R</td><td>0</td></tr>
<tr><td>20</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>19</td><td>RFDF</td><td>Receive FIFO Data Format (async./isochr.)
’0’: No additional status information stored.
’1’: Status byte is stored after each data byte in the receive buffer:
<table>
<tr><td>15</td><td>14</td><td>13…9</td><td>8</td><td>7……0</td></tr>
<tr><td>parity error</td><td>frame error</td><td>reserved</td><td>parity bit</td><td>data byte</td></tr>
</table>
Note: RFDF value is only evaluated while Receiver Reset ‘RRES’ is active.</td><td>R/W</td><td>0</td></tr>
<tr><td>18</td><td>RFTH</td><td>Receive FIFO Threshold
’0’: Maximum PCI bus write burst length for receive data is set to 15 DWORDs (Receive data is written/forwarded to the receive buffer in host memory when the gathered receive data amount is >= 15 DWords or when a Frame/Block-End has been detected).
’1’: Maximum PCI bus write burst length for receive data is set to 1 DWORD (Receive data is written/forwarded to the receive buffer in host memory when the gathered receive data amount is >= 1 DWord or when a Frame/Block-End has been detected).</td><td>R/W</td><td>0</td></tr>
<tr><td>17:4</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>3</td><td>ITF</td><td>Interframe Time Fill (hdlc)
This bit selects the idle state of the transmit pin TxD:
’0’: Continuous logical ’1’ is sent during idle phase.
’1’: Continuous flag sequences are sent (’01111110’ flag pattern).</td><td>R/W</td><td>0</td></tr>
<tr><td>2:1</td><td>-</td><td>Reserved (0 for reads)</td><td>R</td><td>0</td></tr>
<tr><td>0</td><td>XCRC</td><td>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.</td><td>R/W</td><td>0</td></tr>
</table>

Table 6-28 : Channel Configuration Register 2

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

There is a dedicated Baud Rate Generator for each serial channel.

Bit	Symbol	Description	Access	Reset Value
31	BRGM	BRG Mode	R/W	0
30:21	-	Reserved (0 for reads)	R	0
20:0	BRD	Baud Rate Divisor	R/W	0

Table 6-29 : Baud Rate Register

These values determine the divisor of the channel's baud rate generator.

The baud rate generator input clock f_{in} depends on the selected clock source (see also chapters "Clock Sources" and "ACR - Additional Configuration Register").

The resulting output frequency of the baud rate generator is:

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

The divisor k can be set in 2 ways, determined by BRR[31].

When BRR[31] = 0, k is calculated as:

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

with N (BRR[5:0]) = 0..63 and M (BRR[11:8]) = 0..15

The alternative is to set (k – 1) directly as a 21-bit wide value, when BRR[31] = 1:

k = BRR[20:0] + 1 (respectively BRR[20:0] = k – 1)

For Standard Asynchronous serial mode as well as for using the DPLL as a clock source, the baud rate generator output clock shall be set to 16 times higher than the expected/desired data rate.

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

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

Table 6-30 : Termination Character Register

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

Bit	Symbol	Description	Access	Reset Value
31:19	-	‘0’: Interrupt is NOT masked, Interrupt Status is generated and Interrupt is generated ‘1’: Interrupt is Masked, Interrupt Status is generated but no Interrupts is generated See Interrupt Status Register for interrupt bit description. Note: CSC and CDSC are not supported by the TMPE863 and should be kept as ‘1’ (masked).	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

Table 6-31 : Interrupt Mask Register

Unused interrupts shall be masked to avoid unwanted behavior.

Especially CSC and CDSC should be masked because CTS and CD inputs are not supported (unconnected).

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

Bit	Symbol	Description	Access	Reset Value
31:19	-	Reserved (0 for reads)	R	0
18	ALLS	ALL Sent Interrupt HDLC Mode: This bit is set to '1' if the last bit of the current HDLC frame is sent out via pin TxD, 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.	R/C	0
17	-	Reserved (0 for reads)	R	0
16 (hdlc)	XDU	Transmit Data Underrun Interrupt (hdlc) 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. *Note: The transmitter is stopped if this condition occurs and needs to be reset via command bit 'XRES' in register CMDR.*	R/C	0
16 (asyn)	-	Reserved (0 for reads)	R	0
15	-	Reserved (0 for reads)	R	0
14	CSC	CTS# Status Change Interrupt Not supported for TMPE863.	R/C	0
13:10	-	Reserved (0 for reads)	R	0
9 (asyn)	BRK	Break Interrupt (async./isochr.) 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.	R/C	0
8 (asyn)	BRKT	Break Terminated Interrupt (async./isochr.) This bit is set to '1', if a previously detected break condition on the receive line is terminated by a low to high transition.	R/C	0
7 (asyn)	TCD	Termination Character Detected Interrupt (async./isochr.) 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.	R/C	0
6 (asyn)	TIME	Time Out Interrupt (async./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

Bit	Symbol	Description	Access	Reset Value
5 (asyn)	PERR	Parity Error Interrupt (async./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) (hdlc)	FERR	Frame Error Interrupt (async./isochr., hdlc) 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. In HDLC mode, this bit indicates Frame CRC errors.	R/C	0
9:5 (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 Not supported for TMPE863.	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

Table 6-32 : Interrupt Status Register

Unmasked interrupt status bits in the Interrupt Status Register are automatically cleared when the interrupt status word has been logged for DMA transfer to the corresponding SCC interrupt queue in host memory.

Typically, masked interrupt status bits are not visible in the interrupt status register. Masked interrupt status bits will not generate an interrupt and will not be transferred to an interrupt queue in host memory. Masked interrupt status bits are visible in the interrupt status register if the CCR0.VIS bit is set to '1'. To clear such status bits, the host CPU must write a '1' to the corresponding ISR bit.

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

Bit	Symbol	Description	Access	Reset Value
31:22	-	Reserved (0 for reads)	R	0
21	STXFD	Reduced SCC TX FIFO Depth '0': SCC TX FIFO Depth is 16 Byte '1': SCC TX FIFO Depth is 4 Byte	R/W	0
20	-	Reserved	R/W	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) *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.*	R/W	0
18:16	-	Reserved	R/W	0
15	DCMRST	Reset the Clock Multiplier '0': Clock Multiplier is running '1': Clock Multiplier is held in Reset *Note: The Clock Multiplier should always be reset after changing its input clock*	R/W	0
14	X4MULT	Multiply BRG Input Clock x4 '0': BRG input clock is not multiplied by 4 '1': BRG input clock is multiplied by 4 *Note: The input frequency range of the x4 clock multiplier is 10 MHz to 24 MHz. These values must never be exceeded to ensure proper function of the clock multiplier.*	R/W	0
13	RXCINV	Invert RxC '0': No inverting of RxC (the controller samples receive data bits with the falling RxC edge) '1': RxC is inverted (the controller samples receive data bits with the rising RxC edge)	R/W	0
12	TXCINV	Invert TxC '0': No inverting of TxC (the controller generates transmit data bits with the rising TxC edge) '1': TxC is inverted (the controller generates transmit data bits with the falling TxC edge)	R/W	0

Bit	Symbol	Description	Access	Reset Value
11:10	RCS	Receiver Clock Source '00': BRG Output Clock '01': external RxC Input signal '10': DPLL Receive Clock '11': reserved *Note: Typically option '10' is used for FM0, FM1 or Manchester Encoding.*	R/W	00
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 Transmit Clock '100': BRG Output Clock divided by 16 others: reserved *Note: Option '100' is intended for BRG clock based FM0, FM1 or Manchester Encoding (not for Asynchronous protocol).*	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	-	Reserved	R/W	0
2:0	MODE	Transceiver Mode (M2:M0) '001': EIA-422/485 '111': No Cable (High-Impedance) Others: Reserved	R/W	111

Table 6-33 : Additional Configuration Register

The input frequency range of the x4 clock multiplier is 10 MHz to 24 MHz. These values must never be exceeded to ensure proper function of the clock multiplier.

7 Configuration Hints

7.1 Endian Mode

The TMPE863 expects all accesses by the host and all data structures in the host RAM to be in 'Little Endian' format.

Transmit data bytes are fetched & sent from lower to higher byte addresses. Byte bits are sent from Bit 0 (LSb) to Bit 7 (MSb).

Received data bytes are stored from lower to higher byte addresses. Byte bits are gathered from Bit 0 (LSb) to Bit 7 (MSb).

7.2 Configuration EEPROM

An on-board I2C EEPROM contains configuration information data for the software driver and is accessible by the GCTL register (Offset 0xF8).

I2C EEPROM: Type 24C04, I2C Address 1010000, Size 4 Kbit (= 512 Bytes).

The stored information is half-word (16 bit) wise. The EEPROM address map is shown byte (8 bit) wise.

Byte Offset	Description	Value TMPE863-10R
0x00, 0x01	Vendor ID (0x1498 = TEWS Technologies)	0x14, 0x98
0x02, 0x03	Device ID (0xA35F = TMPE863)	0xA3, 0x5F
0x04, 0x05	Subsystem Vendor ID (0x1498 = TEWS Technologies)	0x14, 0x98
0x06, 0x07	Subsystem ID (0xA00A = TMPE Form Factor, Variant -10R)	0xA0, 0x0A
0x08, 0x09	Module Version (V1.0 = 0x0100)	0x01, 0x00
0x0A, 0x0B	Module Revision (0x00 = Rev. A)	0x00, 0x00
0x0C, 0x0D	Module Variant (-10R = 0x0A)	0x00, 0x0A
0x0E, 0x0F	EEPROM Revision	0x00, 0x01
0x10, 0x11	Clock Source 1 Frequency (High)	0x00, 0xE1
0x12, 0x13	Clock Source 1 Frequency (Low)	0x00, 0x00
0x14, 0x15	Clock Source 2 Frequency (High)	0x01, 0x6E
0x16, 0x17	Clock Source 2 Frequency (Low)	0x36, 0x00
0x18, 0x19	Clock Source 3 Frequency (High)	0x00, 0x98
0x1A, 0x1B	Clock Source 3 Frequency (Low)	0x96, 0x80
0x1C - 0x1F	Reserved (2x)	0x00, 0x00
0x20, 0x21	RS232 Channels	0x00, 0x00

0x22, 0x23	RS422 Channels	0x00, 0x07
0x24 - 0x29	Reserved (3x)	0x00, 0x00
0x2A, 0x2B	Multi-Protocol Channels	0x00, 0x00
0x2C - 0x3D	Reserved (9x)	0x00, 0x00
0x3E, 0x3F	Programmable Interfaces	0x00, 0x07
0x40, 0x41	Max Data Rate RS232 (High)	0x00, 0x00
0x42, 0x43	Max Data Rate ES232 (Low)	0x00, 0x00
0x44, 0x45	Max Data Rate RS422 (High)	0x00, 0x98
0x46, 0x47	Max Data Rate ES422 (Low)	0x96, 0x80
0x48- 0x5F	Reserved (12x)	0x00, 0x00
0x60, 0x61	RxD & TxD	0x00, 0x07
0x62, 0x63	RTS & CTS & CD	0x00, 0x00
0x64, 0x65	Full Modem	0x00, 0x00
0x66 - 0xBF	Reserved (45x)	0x00, 0x00
0xC0 - 0xFF	User Programmable Space (optional, 64 Byte useable)	0x00, 0x00

Table 7-1 : Configuration EEPROM Map

Byte Offsets 0x00 - 0xBF are factory programmed and reserved for configuration information for the software driver and **must not be overwritten!**

For compatibility reasons, byte offsets 0x100 - 0x1FF are not used.

Note that the TMPE863 does not feature any RS232 channels or RTS/CTS/CD line signals.

For the physical interfaces: Bit 0 represents channel 0 (the first channel), Bit 1 represents channel 1 and Bit 2 represents channel 2 (the third channel). The appropriate bit is set to '1' for each SCC channel attached to the physical interface represented by the data word. E.g. 0x07 means that all three SCC channels are attached to the appropriate physical interface.

Frequency rate is in Hz. E.g. Clock Source 3 value 0x00989680 means 10 MHz.

Data rate is in bit/s. E.g. Max Data Rate 0x00989680 means 10 Mbit/s.

7.3 Differential Line Wiring & Termination

The TMPE863 TXC and TXD differential transmitters are intended for a point-to-point connection to an external RS422 or RS485 receiver. The external receivers should provide differential line termination (in the range of 100 … 120 Ohm).

The TMPE863 RXC and RXD differential receivers are intended for a point-to-point connection to an external RS422 or RS485 transmitter. The TMPE863 receivers feature on-board differential line termination.

7.4 Transmit & Receive Clock Polarity

By default, transmit data is generated by the rising edge of the transmit clock and receive data is sampled by the falling edge of the receive clock. However, the transmit and receive clock polarity is programmable.

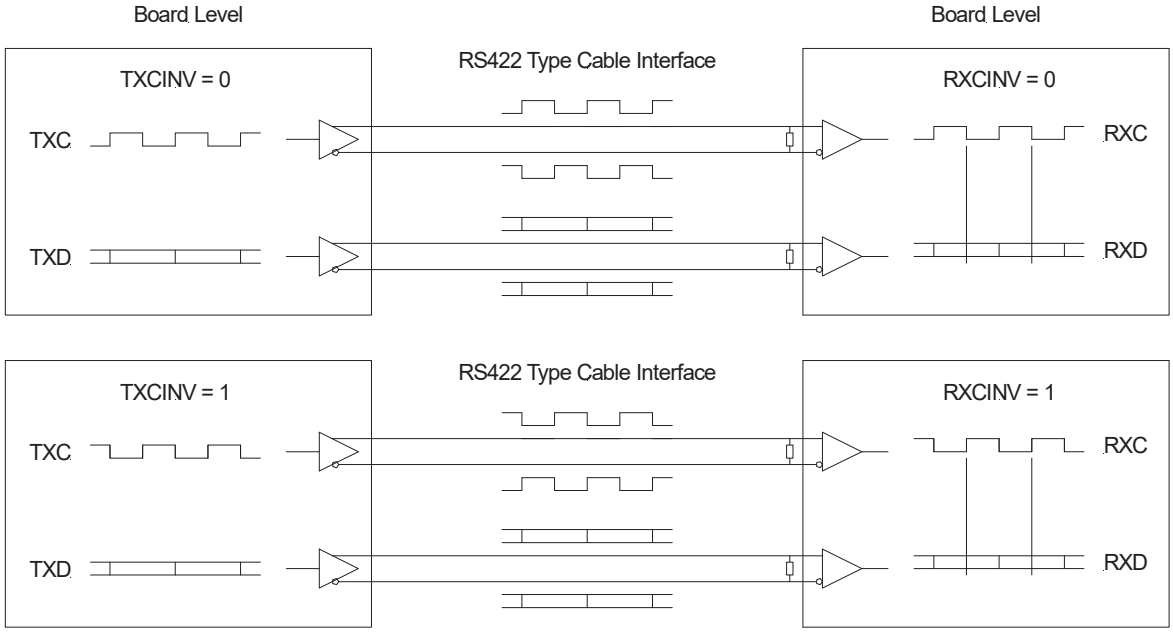
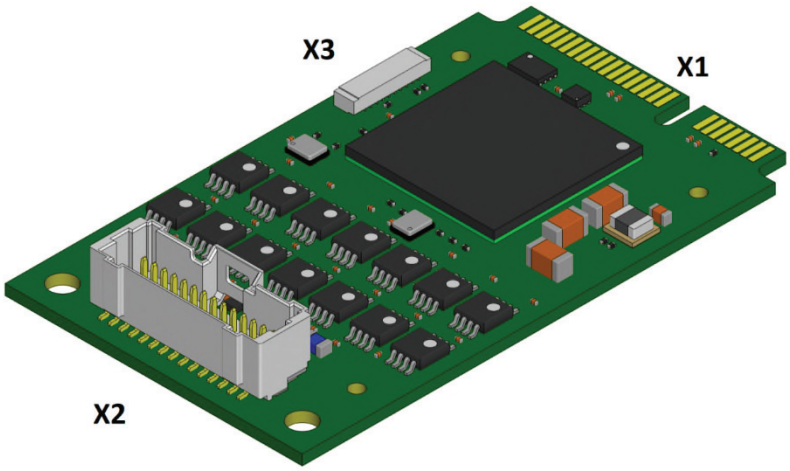


Figure 7-1 : Transmit & Receive Clock Polarity

8 I/O Connectors

This chapter provides information about user accessible on-board connectors.

8.1 Overview



X1	System Connector
X2	I/O Connector
X3	JTAG Connector

Figure 8-1 : I/O Connector Overview

8.2 Board Connectors

8.2.1 System Connector (X1)

Pin-Count	52
Connector Type	Card-edge
Source & Order Info	none

Table 8-1 : System Connector

Signal names in grey are not used by the card.

Description	Pin		Pin	Description
W_DISABLE2#	51		52	+3.3 Vaux
Reserved	49		50	GND
Reserved	47		48	+1.5 V
Reserved	45		46	LEP_WPAN#
GND	43		44	LEP_WLAN#
+3.3 Vaux	41		42	LEP_WWAN#
+3.3 Vaux	39		40	GND
GND	37		38	USB_D+
GND	35		36	USB_D-
PETp0	33		34	GND
PETn0	31		32	SMB_DATA
GND	29		30	SMB_CLK
GND	27		28	+1.5 V
PERp0	25		26	GND
PERn0	23		24	+3.3 Vaux
GND	21		22	PERST#
UIM_IC_DP	19		20	W_DISABLE1#
UIM_IC_DM	17		18	GND
Mechanical Key				Mechanical Key
GND	15		16	UIM_SPU
REF_CLK+	13		14	UIM_RESET
REF_CLK-	11		12	UIM_CLK
GND	9		10	UIM_DATA
CLKREQ#	7		8	UIM_PWR
COEX2	5		6	+1.5 V
COEX1	3		4	GND
WAKE#	1		2	+3.3 Vaux

Table 8-2 : System Connector Pin Assignment

8.2.2 I/O Connector (X2)

Pin-Count	30
Connector Type	Molex Pico-Clasp, dual row straight header, with lock
Source & Order Info	501190-3017
Mating Part	501189-3010

Table 8-3 : I/O Connector

The I/O connector exceeds the available PCI Express Mini Card components height.

Check carefully if you application provides enough spacing for a TMPE863.

Pin Assignment					
Description		**Pin**		**Pin**	**Description**
GND		1		2	GND
RXD[0]+	Input	3		4	RXD[0]- Input
TXD[0]+	Output	5		6	TXD[0]- Output
RXCLK[0]+	Input	7		8	RXCLK[0]- Input
TXCLK[0]+	In/Out	9		10	TXCLK[0]- In/Out
RXD[1]+	Input	11		12	RXD[1]- Input
TXD[1]+	Output	13		14	TXD[1]- Output
RXCLK[1]+	Input	15		16	RXCLK[1]- Input
TXCLK[1]+	In/Out	17		18	TXCLK[1]- In/Out
RXD[2]+	Input	19		20	RXD[2]- Input
TXD[2]+	Output	21		22	TXD[2]- Output
RXCLK[2]+	Input	23		24	RXCLK[2]- Input
TXCLK[2]+	In/Out	25		26	TXCLK[2]- In/Out
Reserved		27		28	Reserved
GND		29		30	GND

Table 8-4 : I/O Connector Pin Assignment

The TMPE863 TXC and TXD differential transmitters are intended for a point-to-point connection to an external RS422 or RS485 receiver. The external receivers should provide differential line termination (in the range of 100 … 120 Ohm).

The TMPE863 RXC and RXD differential receivers are intended for a point-to-point connection to an external RS422 or RS485 transmitter. The TMPE863 receivers feature on-board differential line termination.

TEWS Technologies provides the following matching optional accessory products (not included):

- TA111-10R 30 pos. 500 mm Pico-Clasp Cable Harness
- TA205-10R 30 pos. Pico-Clasp Terminal Block
- TA309-10R Cable Kit for Modules with Pico-Clasp Connector; Combines TA111-10R and TA205-10R.

8.2.3 JTAG Connector (X3)

Pin-Count	10
Connector Type	JST XRS 10pol 0.6 mm Pitch IDC Connector
Source & Order Info	SM10B-XSRS-ETB
Mating Part	10XSR-36S

Table 8-5 : JTAG Connector

The TMPE863 provides a JTAG connector (Factory Use only).