

# TPMC377

**Conduction Cooled**

**4 Channel Isolated RS232/RS422/RS485**

**Programmable Serial Interface**

Version 1.0

## **User Manual**

Issue 1.0.1

August 2014

## TPMC377-10R

Conduction Cooled 4 Channel Isolated  
RS232/RS422 /RS485 Programmable Serial  
Interface, P14 I/O

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

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

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

### Style Conventions

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

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

Access terms are described as:

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

©2014 by TEWS TECHNOLOGIES GmbH

All trademarks mentioned are property of their respective owners.

| Issue | Description      | Date       |
|-------|------------------|------------|
| 1.0.0 | First Issue      | April 2011 |
| 1.0.1 | General Revision | 2014       |

# Table of Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>PRODUCT DESCRIPTION .....</b>            | <b>6</b>  |
| <b>2</b> | <b>TECHNICAL SPECIFICATION .....</b>        | <b>7</b>  |
| <b>3</b> | <b>LOCAL SPACE ADDRESSING .....</b>         | <b>9</b>  |
| 3.1      | XR17D154 Local Space Configuration.....     | 9         |
| 3.2      | Device Configuration Space .....            | 9         |
| 3.2.1    | UART Register Sets.....                     | 9         |
| 3.2.2    | Device Configuration Registers .....        | 11        |
| 3.2.3    | UART Channel Configuration Registers .....  | 12        |
| <b>4</b> | <b>PCI9030 TARGET CHIP .....</b>            | <b>14</b> |
| 4.1      | PCI Configuration Registers (PCR).....      | 14        |
| 4.2      | Configuration EEPROM .....                  | 15        |
| <b>5</b> | <b>CONFIGURATION HINTS.....</b>             | <b>17</b> |
| 5.1      | CPLD Description.....                       | 17        |
| 5.1.1    | CPLD Address Map .....                      | 17        |
| 5.1.2    | Channel Control Register .....              | 17        |
| 5.2      | CPLD Access.....                            | 18        |
| 5.2.1    | Accessing XR17D154 MPIO Pins.....           | 19        |
| 5.2.2    | CPLD Bus Protocol.....                      | 19        |
| 5.2.2.1  | Write.....                                  | 19        |
| 5.2.2.2  | Read .....                                  | 20        |
| 5.3      | Serial Interface Channel Setup.....         | 21        |
| 5.3.1    | Special Features.....                       | 21        |
| 5.3.1.1  | Auto RS485 Operation.....                   | 21        |
| 5.3.1.2  | RS485 Receiver Control.....                 | 22        |
| 5.3.1.3  | Slew Rate Limiting .....                    | 22        |
| 5.3.1.4  | Low-Power Shutdown.....                     | 22        |
| 5.3.2    | Channel Setup .....                         | 23        |
| 5.4      | RS485/RS422 Configuration Examples.....     | 23        |
| 5.4.1    | RS422 Multidrop .....                       | 23        |
| 5.4.2    | RS422 Full Duplex Point to Point .....      | 24        |
| 5.4.3    | RS485 Full Duplex Point to Point .....      | 24        |
| 5.4.4    | RS485 Half Duplex Point to Point.....       | 24        |
| 5.4.5    | RS485 Full Duplex Multi-point .....         | 25        |
| 5.4.6    | RS485 Half Duplex Multi-point.....          | 25        |
| 5.5      | I/O Electrical Interface .....              | 26        |
| 5.5.1    | ±15kV ESD Protection .....                  | 26        |
| 5.5.2    | RS232 Transceivers .....                    | 26        |
| 5.5.3    | RS485/RS422 Transceivers .....              | 26        |
| 5.5.4    | Termination .....                           | 26        |
| 5.5.5    | Galvanic Isolation.....                     | 26        |
| 5.6      | Block Diagram .....                         | 27        |
| <b>6</b> | <b>PROGRAMMING HINTS.....</b>               | <b>28</b> |
| 6.1      | UART Baud Rate Programming.....             | 28        |
| <b>7</b> | <b>PIN ASSIGNMENT – I/O CONNECTOR .....</b> | <b>30</b> |
| 7.1      | Back I/O PMC Connector (P14).....           | 30        |

## List of Figures

|   |    |
|---|----|
| FIGURE 1-1 : BLOCK DIAGRAM.....                                   | 6  |
| FIGURE 5-1 : CPLD BUS BLOCK DIAGRAM.....                          | 18 |
| FIGURE 5-2 : CPLD BUS WRITE .....                                 | 19 |
| FIGURE 5-3 : CPLD BUS READ.....                                   | 20 |
| FIGURE 5-4 : RS422 MULTIDROP CONFIGURATION .....                  | 23 |
| FIGURE 5-5 : RS422 FULL DUPLEX POINT TO POINT CONFIGURATION.....  | 24 |
| FIGURE 5-6 : RS485 FULL DUPLEX POINT TO POINT CONFIGURATION.....  | 24 |
| FIGURE 5-7 : RS485 HALF DUPLEX POINT TO POINT CONFIGURATION ..... | 24 |
| FIGURE 5-8 : RS485 FULL DUPLEX MULTI-POINT CONFIGURATION .....    | 25 |
| FIGURE 5-9 : RS485 HALF DUPLEX MULTI-POINT CONFIGURATION .....    | 25 |
| FIGURE 5-10 : I/O BLOCK DIAGRAM.....                              | 27 |

## List of Tables

|   |    |
|---|----|
| TABLE 2-1 : TECHNICAL SPECIFICATION.....                      | 8  |
| TABLE 3-1 : XR17D154 LOCAL SPACE CONFIGURATION.....           | 9  |
| TABLE 3-2 : DEVICE CONFIGURATION SPACE .....                  | 9  |
| TABLE 3-3 : UART REGISTER SET OFFSET .....                    | 9  |
| TABLE 3-4 : UART REGISTER SET.....                            | 10 |
| TABLE 3-5 : DEVICE CONFIGURATION REGISTERS .....              | 11 |
| TABLE 3-6 : UART CHANNEL CONFIGURATION REGISTERS.....         | 12 |
| TABLE 4-1 : PCI HEADER .....                                  | 14 |
| TABLE 4-2 : CONFIGURATION EEPROM TPMC377-10R.....             | 15 |
| TABLE 4-3 : PHYSICAL CONFIGURATION EEPROM DATA .....          | 16 |
| TABLE 5-1 : CPLD REGISTER ADDRESS MAP.....                    | 17 |
| TABLE 5-2 : CHANNEL CONTROL REGISTER.....                     | 18 |
| TABLE 5-3 : MPIO PINS .....                                   | 18 |
| TABLE 5-4 : MPIO DEVICE CONFIGURATION REGISTERS.....          | 19 |
| TABLE 5-5 : SERIAL CHANNEL SETUP.....                         | 23 |
| TABLE 6-1 : UART BAUD RATE PROGRAMMING .....                  | 28 |
| TABLE 7-1 : PIN ASSIGNMENT BACK I/O PMC CONNECTOR (P14) ..... | 30 |

# 1 Product Description

The TPMC377 is a conduction cooled single-width 32 bit PMC module offering 4 channels of high performance RS232/RS422/RS485 programmable asynchronous serial interface with P14 I/O. Each of the four channels are isolated from the system and against each other by digital isolator and on board integrated DC/DC converter.

The serial channels can be individually programmed to operate as RS232, RS422 or RS485 full duplex / half duplex interface. In addition programmable termination is provided for the RS422/RS485 interfaces. After power-up all serial I/O lines are in a high impedance state.

Each RS232 channel supports Rx/D, Tx/D, RTS, CTS and GND. RS422 and RS485 full duplex support a four wire interface (RX+, RX-, TX+, TX-) plus ground (GND). RS485 half duplex supports a two wire interface (DX+, DX-) plus ground (GND).

Each channel has 64 byte transmit and receive FIFOs to significantly reduce the overhead required to provide data to and get data from the transmitters and receivers. The FIFO trigger levels are programmable and the baud rate is individually programmable up to 921.6 kbps for RS232 channels and 5.5296 Mbps for RS422/RS485 channels. The UART offers readable FIFO levels.

All channels generate interrupts on PCI interrupt INTA. For fast interrupt source detection the UART provides a special Global Interrupt Source Register.

All serial channels use ESD protected transceivers. ESD protection is up to  $\pm 15\text{KV}$ .

The TPMC377 can operate with 3.3V and 5.0V PCI I/O signaling voltage.

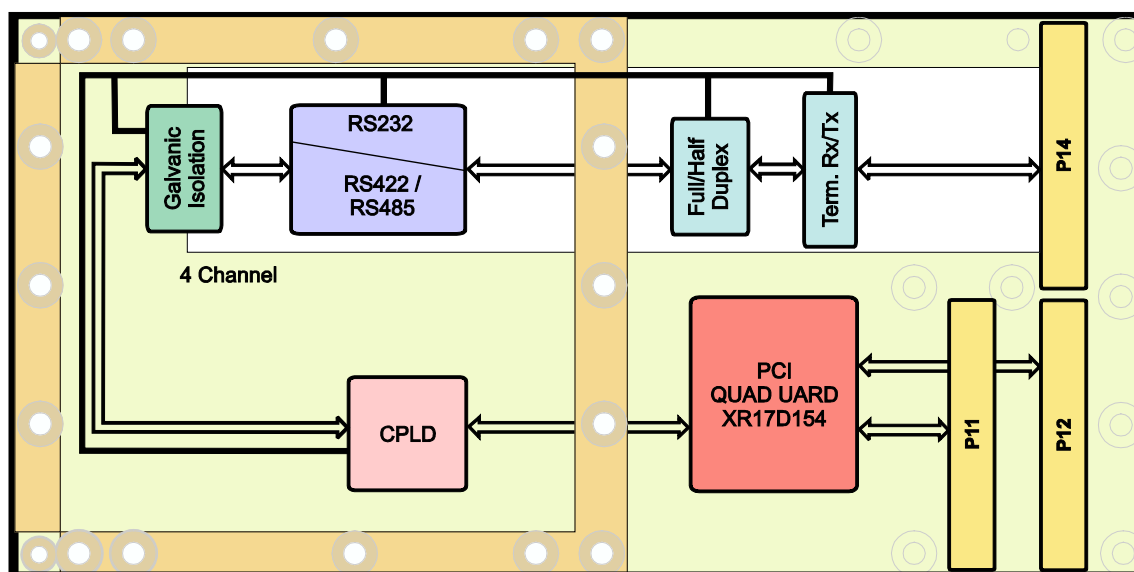


Figure 1-1 : Block Diagram

## 2 Technical Specification

|                            |   |                       |                           |
|----------------------------|---|-----------------------|---------------------------|
| PMC Interface              |   |                       |                           |
| Mechanical Interface       | Conduction Cooled PCI Mezzanine Card (PMC) Interface<br>ANSI/VITA 20-2001 (R2005)<br>Single Size                |                       |                           |
| Electrical Interface       | PCI Rev. 2.3 compliant<br>33 MHz / 32 bit PCI<br>3.3V and 5V PCI Signaling Voltage                              |                       |                           |
| On Board Devices           |   |                       |                           |
| PCI Target Chip            | XR17D154 (Exar)   |                       |                           |
| Quad UART                  | XR17D154 (Exar)   |                       |                           |
| Transceiver                | MAX3161E  |                       |                           |
| Galvanic Isolation         | ADuM5401  |                       |                           |
| I/O Interface              |   |                       |                           |
| Interface Type             | Asynchronous serial interface   |                       |                           |
| Number of Channels         | 4   |                       |                           |
| Physical Interface         | Software selectable RS232, RS422, RS485 full duplex, RS485 half duplex  |                       |                           |
| Serial Channel I/O Signals | RS232: TxD, RxD, RTS, CTS, GND<br>RS422/RS485 Full Duplex: TxD+/-, RxD+/-, GND<br>RS485 Half Duplex: Dx+/-, GND |                       |                           |
| Termination                | Software selectable 120Ω  |                       |                           |
| Programmable Baud Rates    | RS232: up to 921.6 kbps<br>RS422: up to 5.5296 Mbps   |                       |                           |
| ESD Protection             | ±15kV - Human Body Model  |                       |                           |
| I/O Connector              | PMC P14 I/O (64 pin Mezzanine Connector)  |                       |                           |
| Physical Data              |   |                       |                           |
| Power Requirements         | Mode  | I <sub>5,0</sub> [mA] | I <sub>3,3</sub> [mA]     |
|                            | Shutdown, no load   | 200                   | not used<br>by<br>TPMC377 |
|                            | RS232, no load  | 235                   |                           |
|                            | RS422, no load  | 270                   |                           |
|                            | RS485 FD Master, no load  | 700                   |                           |
|                            | RS485 HD, no load   | 285                   |                           |
|                            |   |                       |                           |
|                            | Mode / loopback to TPMC377<br>(Ch.0 to Ch.1, ... ,Ch.6 to Ch.7)   | I <sub>5,0</sub> [mA] | I <sub>3,3</sub> [mA]     |
|                            | RS232, loopback   | 320                   | not used<br>by<br>TPMC377 |
|                            | RS422, loopback   | 455                   |                           |
|                            | RS485 FD Master, loopback   | 950                   |                           |
|                            | RS485 HD, loopback  | 250                   |                           |
|                            | Temperature Range   | Operating             | -40 °C to +85 °C          |
| Storage                    |   | -40°C to +100°C       |                           |

|                 |  |
|-----------------|--|
| <b>MTBF</b>     | TPMC377-10R: 805000 h<br>MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G <sub>B</sub> 20°C.<br>The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation. |
| <b>Humidity</b> | 5 – 95 % non-condensing  |
| <b>Weight</b>   | 55 g   |

Table 2-1 : Technical Specification



## 3 Local Space Addressing

### 3.1 XR17D154 Local Space Configuration

The local on board addressable regions are accessed from the PCI side by using the XR17D154 local space.

| XR17D154 PCI Base Address<br>(Offset in PCI Configuration Space) | PCI Space Mapping | Size (Byte) | Port Width (Bit) | Endian Mode | Description                |
|--|-------------------|-------------|------------------|-------------|----------------------------|
| 0 (0x10)   | MEM               | 2048        | 32               | BIG         | Device Configuration Space |

Table 3-1 : XR17D154 Local Space Configuration

### 3.2 Device Configuration Space

**PCI Base Address: XR17D154 PCI Base Address 0 (Offset 0x10 in PCI Configuration Space).**

The TPMC377 uses the Exar XR17D154 Quad UART to provide and control the 4 channels.

| Device Configuration Space Content | PCI Address                             | Size (Bit) |
|------------------------------------|---|------------|
| UART 0 Register Set                | PCI Base Address 0 + (0x0000 to 0x007F) | 32         |
| Device Configuration Registers     | PCI Base Address 0 + (0x0080 to 0x009F) | 32         |
| UART 0 Register Set                | PCI Base Address 0 + (0x0100 to 0x01FF) | 32         |
| UART 1 Register Set                | PCI Base Address 0 + (0x0200 to 0x03FF) | 32         |
| UART 2 Register Set                | PCI Base Address 0 + (0x0400 to 0x05FF) | 32         |
| UART 3 Register Set                | PCI Base Address 0 + (0x0600 to 0x07FF) | 32         |

Table 3-2 : Device Configuration Space

All registers can be accessed in 8, 16 or 32 bit width with exception to one special case: When reading the receive data together with its LSR register content, the host must read them in 16 or 32 bits format in order to maintain integrity of the data byte with its associated error flags.

#### 3.2.1 UART Register Sets

The Device Configuration Space provides a register set for each of the 4 UARTs.

| UART Register Set | Register Set Offset |
|-------------------|---------------------|
| Serial Channel 0  | 0x0000              |
| Serial Channel 1  | 0x0200              |
| Serial Channel 2  | 0x0400              |
| Serial Channel 3  | 0x0600              |

Table 3-3 : UART Register Set Offset

| Offset Address | Description  | Access | Data Width |
|----------------|--|--------|------------|
| 0x000 – 0x00F  | UART Channel Configuration Registers<br>First 8 registers are 16550 compatible | R/W    | 8, 16, 32  |
| 0x010 – 0x07F  | Reserved   | -      | -          |
| 0x080 – 0x093  | Channel 0: Device Configuration Registers<br>All other channels: Reserved      | R/W    | 8, 16, 32  |
| 0x094 – 0x0FF  | Reserved   | -      | -          |
| 0x100          | Read FIFO – 64 bytes of RX FIFO data   | R      | 8, 16, 32  |
|                | Write FIFO – 64 bytes of TX FIFO data  | W      | 8, 16, 32  |
| 0x140 – 0x17F  | Reserved   | -      | -          |
| 0x180 – 0x1FF  | Read FIFO with errors –<br>64 bytes of RX FIFO data + LSR                      | R      | 16, 32     |

Table 3-4 : UART Register Set

## 3.2.2 Device Configuration Registers

The Device Configuration Registers control general operating conditions and monitor the status of various functions. This includes a 16 bit general purpose counter, multipurpose input/outputs (not supported by the TPMC377), sleep mode, soft-reset and device identification and revision. They are embedded inside the UART 0 Register Set.

| Address | Register     | Description                                | Access | Reset Value |
|---------|--------------|--|--------|-------------|
| 0x080   | INT0 [7:0]   | Channel Interrupt Indicator                | R      | 0x00        |
| 0x081   | INT1 [15:8]  | Interrupt Source Details                   | R      | 0x00        |
| 0x082   | INT2 [23:16] |  | R      | 0x00        |
| 0x083   | INT3 [31:24] |  | R      | 0x00        |
| 0x084   | TIMERCNTL    | Timer Control Register                     | R/W    | 0x00        |
| 0x085   | TIMER        | Reserved                                   | -      | 0x00        |
| 0x086   | TIMERLSB     | Programmable Timer Value                   | R/W    | 0x00        |
| 0x087   | TIMERMSB     |  | R/W    | 0x00        |
| 0x088   | 8XMODE       | Sampling Rate Select                       | R/W    | 0x00        |
| 0x089   | REGA         | Reserved                                   | -      | 0x00        |
| 0x08A   | RESET        | UART Reset                                 | W      | 0x00        |
| 0x08B   | SLEEP        | UART Sleep Mode Enable                     | R/W    | 0x00        |
| 0x08C   | DREV         | Device Revision                            | R      | 0x01        |
| 0x08D   | DVID         | Device Identification                      | R      | 0x28        |
| 0x08E   | REGB         | Simultaneous UART Write & EEPROM Interface | W      | 0x00        |
| 0x08F   | MPIOINT      | MPIO Interrupt Mask                        | R/W    | 0x00        |
| 0x090   | MPIOLVL      | MPIO Level Control                         | R/W    | 0x00        |
| 0x091   | MPIO3T       | MPIO Output Pin Tri-state Control          | R/W    | 0x00        |
| 0x092   | MPIOINV      | MPIO Input Polarity Select                 | R/W    | 0x00        |
| 0x093   | MPIOSEL      | MPIO Input/Output Select                   | R/W    | 0xFF        |

Table 3-5 : Device Configuration Registers

For a detailed description of the Device Configuration Registers please refer to the XR17D154 data sheet which is available on the Exar website ([www.exar.com](http://www.exar.com)).

### 3.2.3 UART Channel Configuration Registers

Each UART channel has its own set of internal UART configuration registers for its own operation control and status reporting. The following table provides the register offsets within a register set, access types and access control:

| Register Offset    | Comment    | Register  | Access | Reset Value |
|--------------------|------------|---|--------|-------------|
| 16550 Compatible   |            |   |        |             |
| 0x00               | LCR[7] = 0 | RHR – Receive Holding Register                    | R      | 0xXX        |
|                    |            | THR – Transmit Holding Register                   | W      |             |
|                    | LCR[7] = 1 | DLL – Baud Rate Generator Divisor Latch Low       | R/W    | 0xXX        |
| 0x01               | LCR[7] = 0 | IER – Interrupt Enable Register                   | R/W    | 0x00        |
|                    | LCR[7] = 1 | DLM – Baud Rate Generator Divisor Latch High      | R/W    | 0xXX        |
| 0x02               |            | ISR – Interrupt Status Register                   | R      | 0x01        |
|                    |            | FCR – FIFO Control Register                       | W      | 0x00        |
| 0x03               |            | LCR – Line Control Register                       | R/W    | 0x00        |
| 0x04               |            | MCR – Modem Control Register                      | R/W    | 0x00        |
| 0x05               |            | LSR – Line Status Register                        | R      | 0x60        |
|                    |            | Reserved  | W      |             |
| 0x06               |            | MSR – Modem Status Register                       | R      | 0xX0        |
|                    |            | – Auto RS485 Delay (not supported by the TPMC377) | W      |             |
| 0x07               | User Data  | SPR – Scratch Pad Register                        | R/W    | 0xFF        |
| Enhanced Registers |            |   |        |             |
| 0x08               |            | FCTR – Feature Control Register                   | R/W    | 0x00        |
| 0x09               |            | EFR – Enhanced Function Register                  | R/W    | 0x00        |
| 0x0A               |            | TXCNT – Transmit FIFO Level Counter               | R      | 0x00        |
|                    |            | TXTRG – Transmit FIFO Trigger Level               | W      |             |
| 0x0B               |            | RXCNT – Receiver FIFO Level Counter               | R      | 0x00        |
|                    |            | RXTRG – Receiver FIFO Trigger Level               | W      |             |
| 0x0C               |            | Xchar – Xon, Xoff Received Flags                  | R      | 0x00        |
|                    |            | Xoff-1 – Xoff Character 1                         | W      |             |
| 0x0D               |            | Reserved  | R      | 0x00        |
|                    |            | Xoff-2 – Xoff Character 2                         | W      |             |
| 0x0E               |            | Reserved  | R      | 0x00        |
|                    |            | Xon-1 – Xon Character 1                           | W      |             |
| 0x0F               |            | Reserved  | R      | 0x00        |
|                    |            | Xon-2 – Xon Character 2                           | W      |             |

Table 3-6 : UART Channel Configuration Registers

The address for a UART Channel Configuration Register *x* in a UART Register Set for channel *y* is:

**PCI Base Address 0 (PCI Base Address for the UART Register Space)**

**+ UART Register Set Offset for *channel y***

**+ Register Offset for *register x***

Addressing example:

The address for the LCR register of UART channel 5 is:

PCI Base Address            (PCI Base Address for the Device Configuration Space)

+ 0x0A00                    (Offset of the UART register set for serial channel 5)

+ 0x0003                    (Offset of the LCR register within a UART register set)

For a detailed description of the serial channel registers please refer to the XR17D154 data sheet which is available on the Exar website ([www.exar.com](http://www.exar.com)).

## 4 PCI9030 Target Chip

### 4.1 PCI Configuration Registers (PCR)

| PCI CFG Register Address | Write '0' to all unused (Reserved) bits    |             |    |                     |           |                 |   | PCI writeable | Initial Values (Hex Values) |           |
|--------------------------|--|-------------|----|---------------------|-----------|-----------------|---|---------------|-----------------------------|-----------|
|                          | 31   | 24          | 23 | 16                  | 15        | 8               | 7 |               |                             | 0         |
| 0x00                     | Device ID                                  |             |    |                     | Vendor ID |                 |   |               | N                           | 0179 1498 |
| 0x04                     | Status                                     |             |    |                     | Command   |                 |   |               | Y                           | 0080 0000 |
| 0x08                     | Class Code                                 |             |    |                     |           | Revision ID     |   | N             | 070002 ??                   |           |
| 0x0C                     | BIST                                       | Header Type |    | PCI Latency Timer   |           | Cache Line Size |   | N             | 00 00 00 00                 |           |
| 0x10                     | Memory Base Address Register (BAR)         |             |    |                     |           |                 |   | Y             | FFFFFF00                    |           |
| 0x14                     | I/O Base Address Register (Unimplemented)  |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x18                     | Base Address Register 0 (Unimplemented)    |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x1C                     | Base Address Register 1 (Unimplemented)    |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x20                     | Base Address Register 2 (Unimplemented)    |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x24                     | Base Address Register 3 (Unimplemented)    |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x28                     | Reserved                                   |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x2C                     | Subsystem ID                               |             |    | Subsystem Vendor ID |           |                 |   | N             | s.b. 1498                   |           |
| 0x30                     | Expansion ROM Base Address (Unimplemented) |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x34                     | Reserved                                   |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x38                     | Reserved                                   |             |    |                     |           |                 |   | N             | 00000000                    |           |
| 0x3C                     | Max_Lat                                    | Min_Gnt     |    | Interrupt Pin       |           | Interrupt Line  |   | Y[7:0]        | 00 00 01 00                 |           |

Table 4-1 : PCI Header

Device ID: 0x0179 TPMC377  
 Vendor ID: 0x1498 TEWS TECHNOLOGIES  
 Revision ID: XR17D154 silicon revision  
 Subsystem ID: 0x000A -10R  
 Subsystem Vendor ID: 0x1498 TEWS TECHNOLOGIES

## 4.2 Configuration EEPROM

After power-on or PCI reset, the XR17D154 loads initial configuration register data from the on board configuration EEPROM.

The configuration EEPROM contains the following configuration data:

- Vendor ID
- Vendor Device ID
- SubSystem Vendor ID
- SubSystem Device ID

See the XR17D154 Manual for more information.

| Address | Configuration Register | Configuration Register Offset | Value  |
|---------|------------------------|-------------------------------|--------|
| 0x00    | Vendor ID              | 0x02                          | 0x1498 |
| 0x01    | Device ID              | 0x00                          | 0x0179 |
| 0x02    | Subsystem Vendor ID    | 0x2E                          | 0x1498 |
| 0x03    | Subsystem ID           | 0x2C                          | s.b.   |

Table 4-2 : Configuration EEPROM TPMC377-10R

Subsystem ID Value (Offset 0x0C): TPMC377-10R 0x000A

The words following the configuration data contain:

- The module version and revision
- The UART clock frequency in Hz
- The physical interface attached to the serial channels
- The maximal baud rate of the transceivers in bps
- The supported control signals of the serial channels

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

| Address   | Configuration Register                                     | TPMC377-10 |
|-----------|--|------------|
| 0x04      | Module Version   | 0x0100     |
| 0x05      | Module Revision  | 0x0000     |
| 0x06      | EEPROM Revision  | 0x0002     |
| 0x07      | Oscillator Frequency (high)                                | 0x02A3     |
| 0x08      | Oscillator Frequency (low)                                 | 0x0000     |
| 0x09-0x0F | Reserved   | -          |
| 0x10      | RS232 Channels   | 0x000F     |
| 0x11      | RS422 Channels   | 0x000F     |
| 0x12      | TTL Channels   | 0x0000     |
| 0x13      | RS485 Full Duplex Channels                                 | 0x000F     |
| 0x14      | RS485 Half Duplex Channels                                 | 0x000F     |
| 0x15-0x1E | Reserved   | -          |
| 0x1F      | Programmable Interfaces                                    | 0x000F     |
| 0x20      | Max Data Rate RS232 (high)                                 | 0x000F     |
| 0x21      | Max Data Rate RS232 (low)                                  | 0x4240     |
| 0x22      | Max Data Rate RS422 (high)                                 | 0x0098     |
| 0x23      | Max Data Rate RS422 (low)                                  | 0x9680     |
| 0x24      | Max Data Rate TTL (high)                                   | 0x0000     |
| 0x25      | Max Data Rate TTL (low)                                    | 0x0000     |
| 0x26      | Max Data Rate RS485 Full Duplex (high)                     | 0x0098     |
| 0x27      | Max Data Rate RS485 Full Duplex (low)                      | 0x9680     |
| 0x28      | Max Data Rate RS485 Half Duplex (high)                     | 0x0098     |
| 0x29      | Max Data Rate RS485 Half Duplex (low)                      | 0x9680     |
| 0x2A-0x2F | Reserved   | -          |
| 0x30      | RxD & TxD  | 0x000F     |
| 0x31      | RTS & CTS  | 0x0000     |
| 0x32      | Full modem   | 0x0000     |
| 0x33-0x37 | Reserved   | -          |
| 0x38      | Enhanced RTS & CTS<br>(Front or Back I/O only)             | 0x0000     |
| 0x39      | Enhanced Full modem<br>(Front or Back I/O only)            | 0x0000     |
| 0x3A      | Channels with enhanced RTS & CTS<br>Support for RS232 only | 0x000F     |
| 0x3B-0x3F | Reserved   | -          |

Table 4-3 : Physical Configuration EEPROM Data



## 5 Configuration Hints

The TPMC377 physical interfaces of the serial channels are individually software programmable to various interface configurations. For this purpose a CPLD provides a control register for each interface channel.

### 5.1 CPLD Description

The CPLD provides a Channel Control Register for each of the interface channels. Each of the Channel Control Registers is individually addressable. The access to this registers is described in detail in chapter "CPLC Access".

#### 5.1.1 CPLD Address Map

Refer to the following chart for the register addresses.

| Address | Register Name              | Size (Bit) |
|---------|----------------------------|------------|
| 000     | Control Register Channel 0 | 7          |
| 001     | Control Register Channel 1 | 7          |
| 010     | Control Register Channel 2 | 7          |
| 011     | Control Register Channel 3 | 7          |

Table 5-1 : CPLD Register Address Map

#### 5.1.2 Channel Control Register

This register is identical for all channels.

| Bit | Symbol     | Description  | Access | Reset Value |
|-----|------------|--|--------|-------------|
| 6   | SHDN       | Active-Low Shutdown-Control. Drive SHDN high to shut down transmitters and charge pump.<br>'0': Normal operation<br>'1': Shutdown  | R/W    | 1           |
| 5   | SLEW LIMIT | Transmitter Speed-Select. Select slew-rate limiting for RS232 and RS485. Slew-rate limits with a logic-level high.<br>'0': Normal data rate limit (RS232: 1 Mbps; RS485: 10 Mbps)<br>'1': Limit data rate to 250 kbps (both RS232 & RS485) | R/W    | 0           |
| 4   | TTERM      | Transmitter Termination Enable<br>Terminate transmit line with a 120Ω termination resistor<br>'0': Termination inactive<br>'1': Termination active   | R/W    | 0           |
| 3   | RTERM      | Receiver Termination Enable<br>Terminate receive line with a 120Ω termination resistor<br>'0': Termination inactive<br>'1': Termination active   | R/W    | 0           |

|   |                  |   |     |   |
|---|------------------|---|-----|---|
| 2 | RENA             | Auto RS485 Receiver Enable<br>When the Auto RTS Control feature of the XR17D154 is used in half duplex configurations, this bit can be used to inhibit the reception of an echo of the own data transmission<br>'0': Normal operation (receiver is always enabled)<br>'1': Inhibit echo reception (receiver is disabled during data transmission) | R/W | 0 |
| 1 | HDPLX            | Selectable Mode Functionality. Operates in full-duplex mode when low; operates in half-duplex mode when high.<br>'0': Full-duplex<br>'1': Half-Duplex   | R/W | 0 |
| 0 | RS485/<br>RS232# | Selectable Mode Functionality. Operates as RS485 with a logic-level high; operates as RS232 with a logic-level low.<br>'0': RS232<br>'1': RS485   | R/W | 0 |

Table 5-2 : Channel Control Register

## 5.2 CPLD Access

The CPLD is connected to the MPIO pins of the XR17D154 to provide access to the control registers.

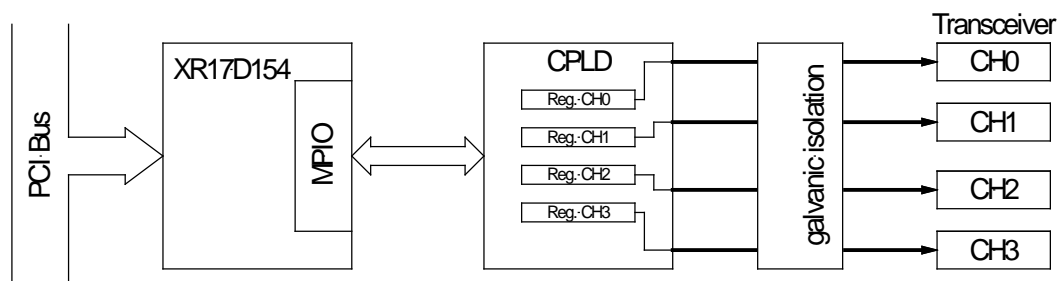


Figure 5-1 : CPLD Bus Block Diagram

The MPIO pins form a simple bus to the CPLD. The following chart gives an overview about the MPIO pin assignment:

| MPIO Pin  | Direction | Function  |
|-----------|-----------|---|
| MPIO[0]   | Output    | CEN – Chip Enable   |
| MPIO[1]   | Output    | R/W# - Read/Write<br>Low: Write to Address<br>High: Read from Address |
| MPIO[2]   | Output    | CLK – Clock   |
| MPIO[5-3] | Output    | ADR – Register Address  |
| MPIO[6]   | Output    | DATAOUT – Serial Data Output  |
| MPIO[7]   | Input     | DATAIN – Serial Data Input  |

Table 5-3 : MPIO Pins

## 5.2.1 Accessing XR17D154 MPIO Pins

The MPIO Registers are accessible at PCI Base Address 0 + Device Configuration Register Offset.

| Address Offset | Register | Description                       | Access | Reset Value |
|----------------|----------|-----------------------------------|--------|-------------|
| 0x08F          | MPIOINT  | MPIO Interrupt Mask               | R/W    | 0x00        |
| 0x090          | MPIOLVL  | MPIO Level Control                | R/W    | 0x00        |
| 0x091          | MPIO3T   | MPIO Output Pin Tri-state Control | R/W    | 0x00        |
| 0x092          | MPIOINV  | MPIO Input Polarity Select        | R/W    | 0x00        |
| 0x093          | MPIOSEL  | MPIO Input/Output Select          | R/W    | 0xFF        |

Table 5-4 : MPIO Device Configuration Registers

MPIOINT, MPIOLVL and MPIO3T must be left at their default values. MPIOSEL must be set to 0x80 to configure MPIO[6-0] pins as outputs and MPIO[7] as input. MPIOLVL sets the output level of the MPIO output pins and is used to write on the CPLD bus.

## 5.2.2 CPLD Bus Protocol

### 5.2.2.1 Write

A CPLD register write access starts with setting CEN to '1'. This resets the CPLD's internal state-machine. Before the first clock pulse is issued, ADR and DATAOUT have to be set; R/W# must be left '0'. The first rising edge of CLK samples the ADR bits and the DATAOUT bit. ADR determines which internal register is accessed. The following 6 rising edges of CLK sample the remaining data bits, additional CLK pulses are ignored. Setting CEN back to '0' completes the access and the configuration of the transceivers will be updated.

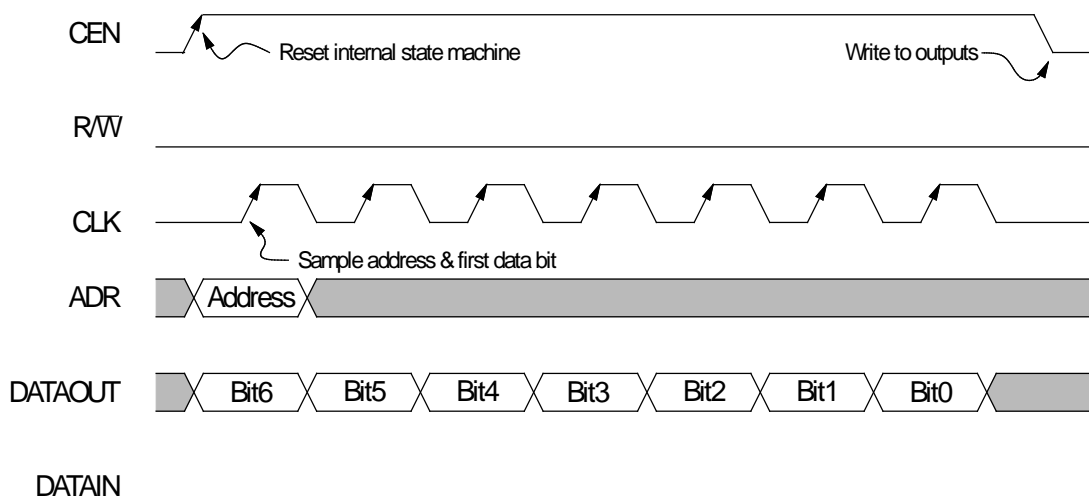


Figure 5-2 : CPLD Bus Write

Example (pseudocode):

Write value 0x05 to CPLD address 0x03

```
define MPIOLVL 0x90;
define MPIOSEL 0x93;
void Write_XR17D154(int address, int value);

Write_XR17D154(MPIOSEL, 0x80);           // Setting up MPIOSEL

Write_XR17D154(MPIOLVL, 0x19);           // CEN = '1', ADR = "011", D(6) = '0'
Write_XR17D154(MPIOLVL, 0x1D);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x01);           // CLK = '0', ADR = "000", D(5) = '0'
Write_XR17D154(MPIOLVL, 0x05);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x01);           // CLK = '0', D(4) = '0'
Write_XR17D154(MPIOLVL, 0x05);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x01);           // CLK = '0', D(3) = '0'
Write_XR17D154(MPIOLVL, 0x05);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x41);           // CLK = '0', D(2) = '1'
Write_XR17D154(MPIOLVL, 0x45);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x01);           // CLK = '0', D(1) = '0'
Write_XR17D154(MPIOLVL, 0x05);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x41);           // CLK = '0', D(0) = '1'
Write_XR17D154(MPIOLVL, 0x45);           // CLK = '1'
Write_XR17D154(MPIOLVL, 0x00);           // CLK = '0', CEN = '0'
```

### 5.2.2.2 Read

A CPLD register read access starts with setting CEN to '1'. This resets the CPLD's internal state-machine. Before the first clock pulse is issued, ADR and R/W# have to be set. The first rising edge of CLK samples the ADR bits and starts the output of the first DATAIN bit. ADR determines which internal register is accessed. The following 6 rising edges of CLK put out the remaining data bits, additional CLK pulses are ignored. Setting CEN back to '0' completes the access.

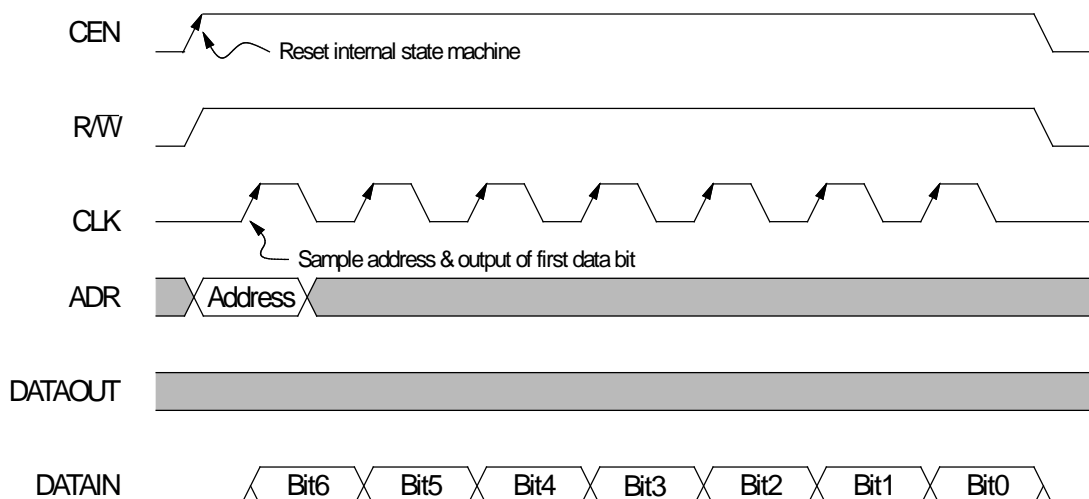


Figure 5-3 : CPLD Bus Read

Example (pseudocode):

Read value 0x05 to CPLD address 0x03

```
define MPIOLVL 0x90;
define MPIOSEL 0x93;
int      Read_XR17D154(int address); // Returns the address' bit 7
int      value

Write_XR17D154(MPIOSEL, 0x80);          // Setting up MPIOSEL

Write_XR17D154(MPIOLVL, 0x1B);          // CEN = '1', ADR = "011", R/W# = '1'
Write_XR17D154(MPIOLVL, 0x1F);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0', ADR = "000"
value |= (Read_XR17D154(MPIOLVL) << 6); // Read D(6)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 5); // Read D(5)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 4); // Read D(4)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 3); // Read D(3)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 2); // Read D(2)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 1); // Read D(1)
Write_XR17D154(MPIOLVL, 0x07);          // CLK = '1'
Write_XR17D154(MPIOLVL, 0x03);          // CLK = '0'
value |= (Read_XR17D154(MPIOLVL) << 0); // Read D(0)
Write_XR17D154(MPIOLVL, 0x00);          // CEN = '0', R/W# = '0'
```

## 5.3 Serial Interface Channel Setup

After power-up all transceivers are in shutdown mode, i.e. the outputs are in tri-state mode. Therefore the serial interfaces must be properly set up before they can be used.

The interfaces can be programmed to following Modes:

- RS232
- RS485/RS422 full-duplex (with optional termination)
- RS485 half-duplex (Master/Slave, with optional termination)

### 5.3.1 Special Features

#### 5.3.1.1 Auto RS485 Operation

In RS485 half duplex applications it is necessary to tristate the driver when it is not active. The XR17D154 provides a special function, the "Auto RS485 Operation" for this purpose. The UART's RTS signal is connected to the driver enable pin of the transceiver. The UART asserts RTS to enable the driver before it starts to send a character and deasserts the RTS signal after a programmable delay after the stop bit of the last transmitted character. The delay optimizes the time needed for the last transmission to reach the farthest station on a long cable network before switching off the line driver.

The Auto RS485 Operation is enabled by FCTR bit 5. The delay is specified in MSR[7:4]

### **5.3.1.2 RS485 Receiver Control**

In RS485 half duplex applications the driver and receiver are connected with each other. To prevent the echo of local data, the receive line can be inhibited for the time the driver is enabled. This is done by activating the “Auto RS485 Receiver Enable” in the Channel Control Register.

When the Auto RS485 Receiver Enable is not activated in a half duplex application, this will result in a kind of loopback mode. This may be done on purpose to monitor the loopback data for errors which would indicate a line contention. When the channel is unconnected, this may also be used as a build in self test.

### **5.3.1.3 Slew Rate Limiting**

The SLEW LIMIT control is used to select the slew-rate limiting of the RS232 transmitters and the RS485/RS422 drivers. With SLEW LIMIT asserted, the RS232 transmitters and the RS485/RS422 driver are slew-rate limited to reduce EMI, resulting in a max data rate of 250kbps. RS232 data rates up to 1Mbps and RS485/RS422 data rates up to 10Mbps are possible when SLEW LIMIT is unasserted. SLEW LIMIT can be changed during operation without interrupting data communications.

### **5.3.1.4 Low-Power Shutdown**

The MAX3161E has a shutdown control input, SHDN. When SHDN is ON, the charge pump and transmitters are shut down and supply current is reduced to 10nA. The RS232 receiver outputs remain active if in RS232 mode. The charge-pump capacitors must be recharged when coming out of shutdown before resuming operation in either RS232 or RS485/RS422 mode.

## 5.3.2 Channel Setup

Each interface channel must be set up in its associated Channel Control Register in the CPLD. Depending on the interface configuration the “Auto RS485 Operation” must be activated in the Feature Control Register in the XR17D154. Refer to the next chapter “RS485/RS422 Configuration Examples” to find out which interface configurations suits your needs.

The following table shows how to program the interfaces to the commonly used modes:

| Bit             | Symbol               | Reset Value                            | RS232   | RS422 Multidrop | RS422 FD | RS485 FD (Master) | RS485 FD Slave | RS485 HD |
|-----------------|----------------------|--|---------|-----------------|----------|-------------------|----------------|----------|
| <b>CPLD</b>     |                      | <b>Channel Control Register</b>        |         |                 |          |                   |                |          |
| 0               | RS485/RS232#         | OFF                                    | OFF     | ON              | ON       | ON                | ON             | ON       |
| 1               | HDPLX                | OFF                                    | OFF     | OFF             | OFF      | OFF               | OFF            | ON       |
| 2               | RENA                 | OFF                                    | OFF     | OFF             | OFF      | OFF               | OFF            | ON       |
| 3               | RTERM                | OFF                                    | OFF (1) | ON (2)          | ON       | ON                | ON (2)         | OFF      |
| 4               | TTERM                | OFF                                    | OFF (1) | OFF             | OFF      | ON                | ON (2)         | ON (2)   |
| 5               | SLEW LIMIT           | OFF                                    | User    | User            | User     | User              | User           | User     |
| 6               | SHDN                 | ON                                     | OFF     | OFF             | OFF      | OFF               | OFF            | OFF      |
| <b>XR17D154</b> |                      | <b>Feature Control Register (FCTR)</b> |         |                 |          |                   |                |          |
| 5               | Auto RS485 Operation | OFF                                    | OFF     | OFF             | OFF      | OFF               | ON             | ON       |

(1) RTERM / TTERM settings are ignored in RS232 mode.

(2) Depends on bus configuration. Terminate only if the transceiver is the end-point of the bus.

Table 5-5 : Serial Channel Setup

## 5.4 RS485/RS422 Configuration Examples

### 5.4.1 RS422 Multidrop

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | OFF   | OFF  | ON*   | OFF   | OFF     |

\* Terminate only if the device is a receiver and the end-point of the bus.

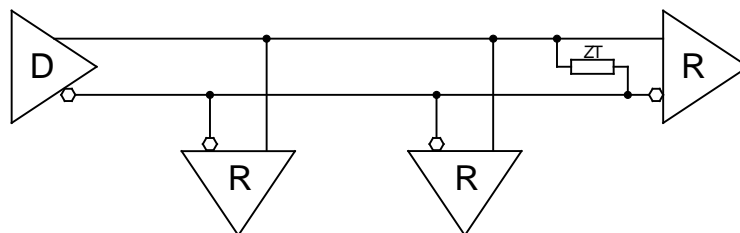


Figure 5-4 : RS422 Multidrop Configuration

## 5.4.2 RS422 Full Duplex Point to Point

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | OFF   | OFF  | ON    | OFF   | OFF     |

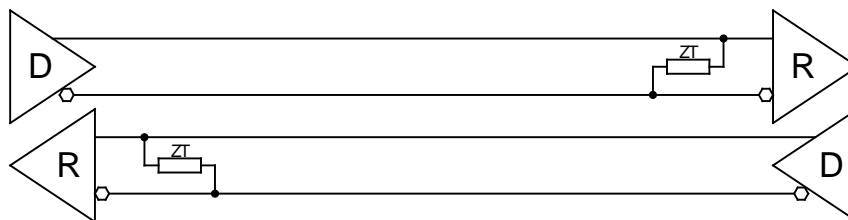


Figure 5-5 : RS422 Full Duplex Point to Point Configuration

## 5.4.3 RS485 Full Duplex Point to Point

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | OFF   | OFF  | ON    | ON    | OFF     |

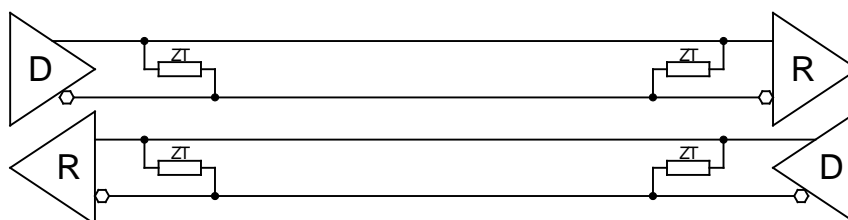


Figure 5-6 : RS485 Full Duplex Point to Point Configuration

## 5.4.4 RS485 Half Duplex Point to Point

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | ON    | ON   | OFF   | ON    | ON      |

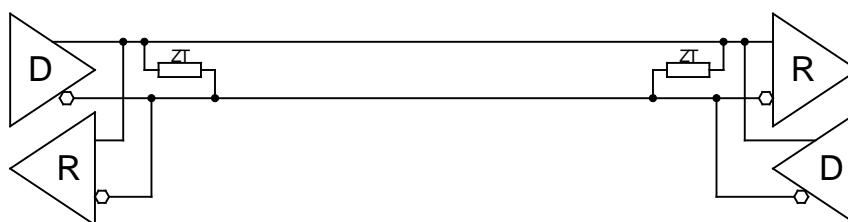


Figure 5-7 : RS485 Half Duplex Point to Point Configuration



## 5.4.5 RS485 Full Duplex Multi-point

Also referred to as “party-line”

**Master**

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | OFF   | OFF  | ON    | ON    | OFF     |

**Slave**

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | OFF   | OFF  | ON*   | ON*   | ON      |

\* Terminate only if the device is the end-point of the bus.

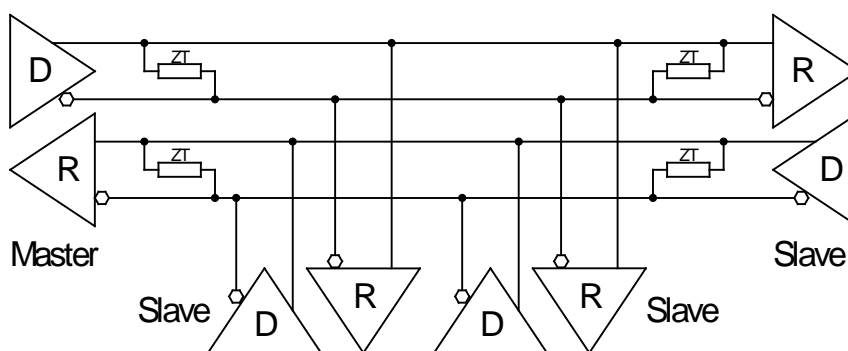


Figure 5-8 : RS485 Full Duplex Multi-Point Configuration

## 5.4.6 RS485 Half Duplex Multi-point

Also referred to as “party-line”

| RS485 | HDPLX | RENA | RTERM | TTERM | FCTR[5] |
|-------|-------|------|-------|-------|---------|
| ON    | ON    | ON   | OFF   | ON*   | ON      |

\* Terminate only if the device is the end-point of the bus.

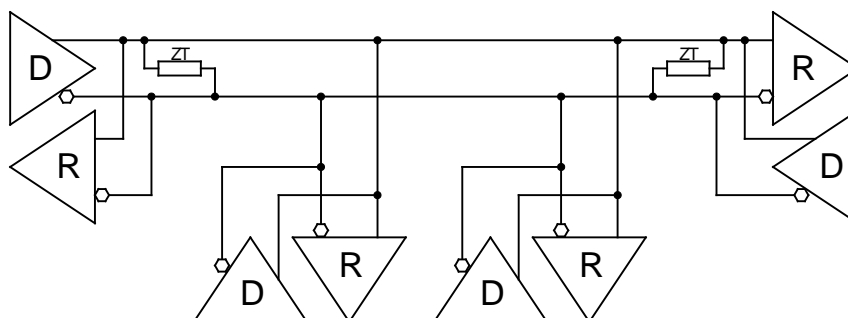


Figure 5-9 : RS485 Half Duplex Multi-Point Configuration

## 5.5 I/O Electrical Interface

### 5.5.1 $\pm 15\text{kV}$ ESD Protection

The receiver inputs and transmitter outputs are characterized for  $\pm 15\text{kV}$  ESD protection using the Human Body Model.

### 5.5.2 RS232 Transceivers

The RS232 transmitters are inverting-level translators that convert CMOS-logic levels to  $\pm 5\text{V}$  EIA/TIA-232-compliant levels. The transmitters are guaranteed at a 250kbps data rate in slew-rate limited mode with worst-case loads of  $3\text{k}\Omega$  in parallel with  $1000\text{pF}$ . Data rates up to 1Mbps can be achieved by not asserting SLEW LIMIT. When powered down or in shutdown, the outputs are high impedance and can be driven to  $\pm 13.2\text{V}$ .

The receivers convert RS232 signals to CMOS-logic output levels. All receivers have inverting outputs that remain active in shutdown. The MAX3161E permit their receiver inputs to be driven to  $\pm 25\text{V}$ . Floating receiver input signals are pulled to ground through internal  $5\text{k}\Omega$  resistors, forcing the outputs to a logic-high.

### 5.5.3 RS485/RS422 Transceivers

The RS485/RS422 transceivers feature fail-safe circuitry that guarantees a logic-high receiver output when the receiver inputs are open or shorted, or when they are connected to a terminated transmission line with all drivers disabled. They also feature selectable reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, allowing error-free data transmission up to 250kbps. The transmitters can operate at speeds up to 10Mbps with the slew-rate limiting disabled. Drivers are short-circuit current limited and thermally limited to protect them against excessive power dissipation. Half-duplex communication is enabled by driving HDPLX high.

### 5.5.4 Termination

The receive and the transmit line can be terminated with a  $120\Omega$  termination resistor. The termination is software selectable.

### 5.5.5 Galvanic Isolation

Each of the four channels are isolated from the system and against each other by digital isolator and on board integrated DC/DC converter.

## 5.6 Block Diagram

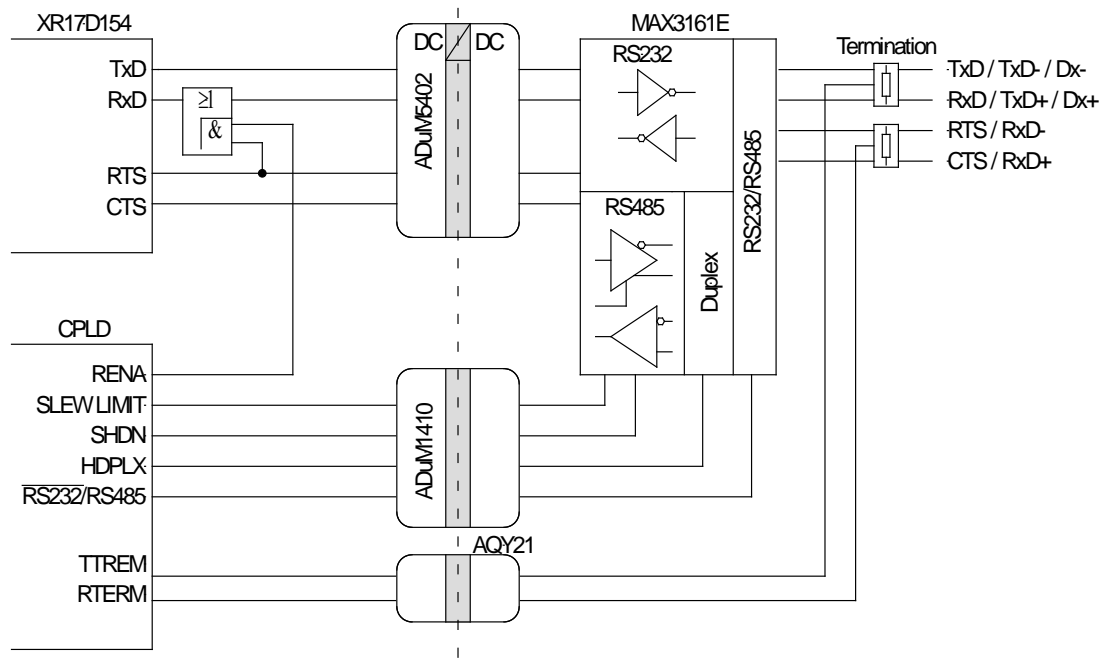


Figure 5-10 : I/O Block Diagram

## 6 Programming Hints

### 6.1 UART Baud Rate Programming

Each of the 4 UART channels of the TPMC377 provides a programmable Baud Rate Generator. The clock of the XR17D154 UART can be divided by any divisor from 1 to  $2^{16} - 1$ . The divisor can be programmed by the UART channel DLM (Divisor MSB) and DLL (Divisor LSB) registers. After a reset bit 7 of the UART channels MCR register defaults to '0' and the divisor value is 0xFFFF.

The basic formula of baud rate programming is:

$$\text{Baud Rate} = \frac{44.2368\text{MHz}}{16 \cdot \text{Divisor} \cdot (1 + 3 \cdot \text{MCR}[7])}$$

Examples for standard baud rates are given in following chart:

| Baud Rate<br>MCR[7] = 0 | Baud Rate<br>MCR[7] = 1 | Divisor | DLM<br>Value | DLL<br>Value |
|-------------------------|-------------------------|---------|--------------|--------------|
| 400                     | 100                     | 0x1B00  | 0x1B         | 0x00         |
| 600                     | 150                     | 0x1200  | 0x12         | 0x00         |
| 1200                    | 300                     | 0x0900  | 0x09         | 0x00         |
| 2400                    | 600                     | 0x0480  | 0x04         | 0x80         |
| 4800                    | 1200                    | 0x0240  | 0x02         | 0x40         |
| 9600                    | 2400                    | 0x0120  | 0x01         | 0x20         |
| 19.2k                   | 4800                    | 0x0090  | 0x00         | 0x90         |
| 38.4k                   | 9600                    | 0x0048  | 0x00         | 0x48         |
| 57.6k                   | 14.4k                   | 0x0030  | 0x00         | 0x30         |
| 115.2k                  | 28.8k                   | 0x0018  | 0x00         | 0x18         |
| 230.4k                  | 57.6k                   | 0x000C  | 0x00         | 0x0C         |
| 460.8k                  | 115.2k                  | 0x0006  | 0x00         | 0x06         |
| 921.6k                  | 230.4k                  | 0x0003  | 0x00         | 0x03         |
| 1382.4k                 | 345.6k                  | 0x0002  | 0x00         | 0x02         |
| 2764.8k                 | 691.2k                  | 0x0001  | 0x00         | 0x01         |

Table 6-1 : UART Baud Rate Programming

To calculate a divisor value for a given baud rate, use following formula:

$$\text{Divisor} = \frac{44.2368\text{MHz}}{16 \cdot \text{Baud Rate} \cdot (1 + 3 \cdot \text{MCR}[7])}$$

The sampling rate for a UART channel can be set to 8x (normal operation is 16x) in the 8XMODE register. Transmit and receive data rates will double by selecting 8x sample rate.

The maximum achievable baud rate is 5.5296 Mbps (Divisor = 0x0001 & 8x sampling rate).

These steps should be used to modify the DLM, DLL registers of an UART channel:

1. Write 0x80 to the LCR register of the UART channel (enable access to the DLM, DLL registers).
2. Program the DLM, DLL registers of the UART channel.
3. Write normal operation byte value to the LCR register of the UART channel.

These steps should be used to modify MCR register bit 7 of an UART channel (set baud rate generator prescaler):

1. Set UART channel EFR register bit 4 to '1' (enable modification of MCR register bits 5-7).
2. Modify UART channel MCR register bit 7.
3. Set UART channel EFR register bit 4 to '0' (latch modified MCR register setting).

**Note that the maximum baud rate for RS232 channel is 921.6 kps. Thus the minimum divisor value for RS232 channels is 0x0003 with MCR[7] = 0.**

## 7 Pin Assignment – I/O Connector

### 7.1 Back I/O PMC Connector (P14)

| Pin | RS232 | RS485/RS422 | Pin | RS232 | RS485/RS422 |
|-----|-------|-------------|-----|-------|-------------|
| 1   | GND_0 |             | 33  | RxD3  | TxD3+/Dx3+  |
| 2   | TxD0  | TxD0-/Dx0-  | 34  | RTS3  | RxD3-       |
| 3   | RxD0  | TxD0+/Dx0+  | 35  | CTS3  | RxD3+       |
| 4   | RTS0  | RxD0-       | 36  | -     | -           |
| 5   | CTS0  | RxD0+       | 37  | -     | -           |
| 6   | -     | -           | 38  | -     | -           |
| 7   | -     | -           | 39  | -     | -           |
| 8   | -     | -           | 40  | -     | -           |
| 9   | -     | -           | 41  | -     | -           |
| 10  | -     | -           | 42  | -     | -           |
| 11  | GND_1 |             | 43  | -     | -           |
| 12  | TxD1  | TxD1-/Dx1-  | 44  | -     | -           |
| 13  | RxD1  | TxD1+/Dx1+  | 45  | -     | -           |
| 14  | RTS1  | RxD1-       | 46  | -     | -           |
| 15  | CTS1  | RxD1+       | 47  | -     | -           |
| 16  | -     | -           | 48  | -     | -           |
| 17  | -     | -           | 49  | -     | -           |
| 18  | -     | -           | 50  | -     | -           |
| 19  | -     | -           | 51  | -     | -           |
| 20  | -     | -           | 52  | -     | -           |
| 21  | GND_2 |             | 53  | -     | -           |
| 22  | TxD2  | TxD2-/Dx2-  | 54  | -     | -           |
| 23  | RxD2  | TxD2+/Dx2+  | 55  | -     | -           |
| 24  | RTS2  | RxD2-       | 56  | -     | -           |
| 25  | CTS2  | RxD2+       | 57  | -     | -           |
| 26  | -     | -           | 58  | -     | -           |
| 27  | -     | -           | 59  | -     | -           |
| 28  | -     | -           | 60  | -     | -           |
| 29  | -     | -           | 61  | -     | -           |
| 30  | -     | -           | 62  | -     | -           |
| 31  | GND_3 |             | 63  | -     | -           |
| 32  | TxD3  | TxD3-/Dx3-  | 64  | -     | -           |

Table 7-1 : Pin Assignment Back I/O PMC Connector (P14)