*The Embedded I/O Company*

TEWS TECHNOLOGIES

# TXMC635

## Reconfigurable FPGA with 48 x TTL IO / 32 x 16 Bit Analog In / 8 x 16 Bit Analog Out

Version 1.0

## User Manual

Issue 1.0.4

February 2020

**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

## TXMC635-10R

48 TTL Front I/O and 64 direct FPGA I/O Lines, 32 x Analog In, 8 x Analog Out,

XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

## TXMC635-20R

48 TTL Front I/O and 64 direct FPGA I/O Lines, 32 x Analog In, 8 x Analog Out,

XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3



### 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 |

| Issue | Description | Date |
|---|---|---|
| 1.0.0 | Initial issue | July 2015 |
| 1.0.1 | Alternative configurations SPI-Flash assembly | May 2017 |
| 1.0.2 | o Analog Input filter added to chapter “Technical Specification”<br>o Chapter 8.1 “Analog Inputs” added<br>o Note about single-ended and differential usability of the analog inputs added to chapter 9.1.3 | April 2018 |
| 1.0.3 | Supplementary description for clock generation. | July 2018 |
| 1.0.4 | Correction of the DAC Interface Output Amplifier type in the Serial DAC Interface chapter. | February 2020 |

# Table of Contents

# List of Figures



# List of Tables

# 1 Product Description

The TXMC635 is a standard single-width Switched Mezzanine Card (XMC) compatible module providing a user configurable XC6SLX45T-2 or XC6SLX100T-2 Xilinx Spartan-6 FPGA.

48 ESD-protected TTL lines provide a flexible digital interface. All I/O lines are individually programmable as input or output. Setting as input sets the I/O line to tri-state and could be used with on-board pull up also as open drain output. Each TTL I/O line has a pull resistor. The pull voltage level is selectable to be either +3.3V, +5V and additionally GND.

8 channels of 16 bit analog outputs allow software selectable output voltage ranges of ±10V, ±10.2564V or ±10.5263V. The output voltage range can be individually set per channel. The conversion time is at most 10 µs and the DAC outputs are routed via operational amplifier in order to protect DAC from damage.

32 ADC input channels can be software configured to operate in single-ended or differential mode with 16 input channels. Each of the 32 channels has a resolution of 16 bit and can work with up to 1 MSPS. The programmable gain amplifier is software configurable and allows a full-scale input voltage range of up to ±24.576V.

For customer specific I/O extension or inter-board communication, the TXMC635-xxR provides 64 FPGA I/Os lines on P14 and 3 FPGA Multi-Gigabit-Transceiver on P16. P14 I/O lines could be configured as 64 single ended LVCMOS33 or as 32 differential LVDS33 interface.

The User FPGA is connected to a 128 Mbytes, 16 bit wide DDR3 SDRAM. The SDRAM-interface uses a hardwired internal Memory Controller Block of the Spartan-6.

The User FPGA is configured by a platform SPI flash or via PCIe download. The flash device is in-system programmable. An in-circuit debugging option is available via a JTAG header for read back and real-time debugging of the FPGA design (using Xilinx “ChipScope”).

The direct configuration via PCIe of the User FPGA is realized by the Configuration FPGA. Configuration data is programmed via 32 bit transfer register to the User FPGA (Spartan6). Data source are XILINX ISE binary files (.bit file or .bin file) which are generated by XILINX ISE Design Software. These binary files consist of header, preamble and configuration data. Only configuration data must be transferred. See also the XILINX User Guide (ug380) “Spartan6 FPGA Configuration” for more information about configuration details and configuration data file formats.

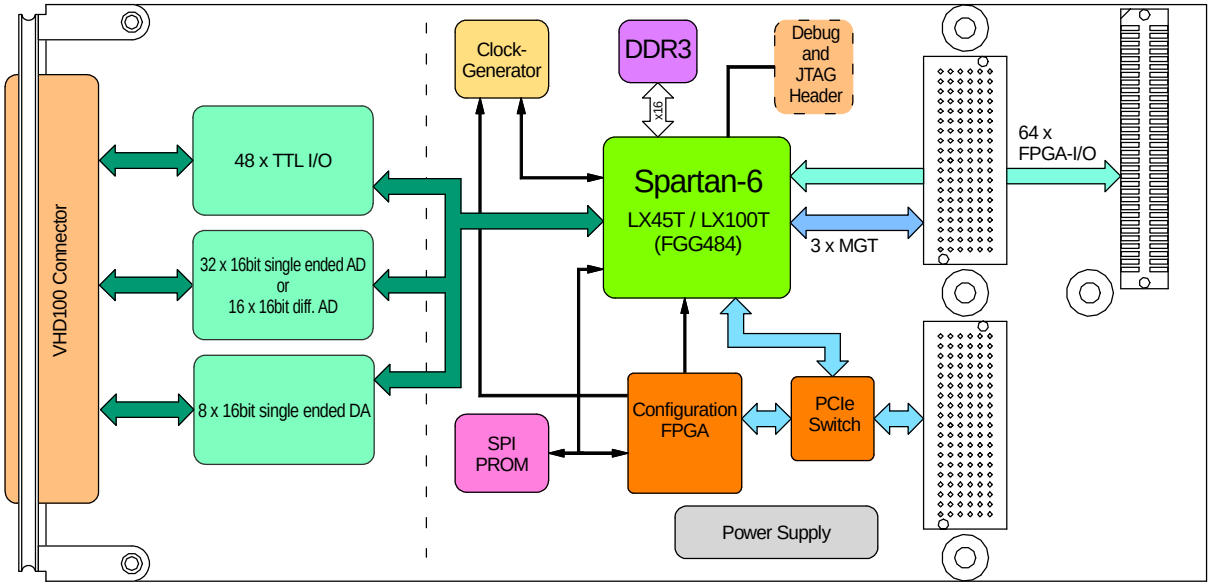



Figure 1-1 : Block Diagram

User applications for the TXMC635 with XC6SLX45T-2 FPGA can be developed using the design software ISE Project Navigator (ISE) and Embedded Development Kit (EDK). IDE versions are 14.7. Licenses for both design tools are required.

TEWS offers a well-documented basic FPGA Example Application design. It includes an .ucf file with all necessary pin assignments and basic timing constraints. The example design covers the main functionalities of the TXMC635. It implements local bus interface to local bridge device, register mapping, DDR3 memory access and basic I/O. It comes as a Xilinx ISE project with source code and as a ready-to-download bit stream.

# 2 Technical Specification

| XMC Interface | |
|---|---|
| **Mechanical Interface** | Switched Mezzanine Card (XMC) Interface confirming to ANSI/VITA 42.0-2008 (Auxiliary Standard)<br>Standard single-width (149mm x 74mm) |
| **Electrical Interface** | PCI Express x1 Link (Base Specification 1.1) compliant interface conforming to ANSI/VITA 42.3-2006 (XMC PCI Express Protocol Layer Standard) |

| On-Board Devices | |
|---|---|
| **PCI Express Switch** | PI7C9X2G404 (Pericom) |
| **PCI Express to PCI Bridge** | XIO2001 (Texas Instruments) |
| **PCI Express Endpoint** | Spartan-6 PCI Express Endpoint Block |
| **User configurable FPGA** | TXMC635-10R: XC6SLX45T-2 (Xilinx)<br>TXMC635-20R: XC6SLX100T-2 (Xilinx) |
| **SPI-Flash** | W25Q32FV (Winbond) or W25Q32JV (Winbond) 32 Mbit (contains TXMC635 FPGA Example) |
| **DDR3 RAM** | MT41J64M16 or MT41K64M16 (Micron) 64 Meg x 16 Bit |
| **Board Configuration FPGA** | LCMXO2-2000HC (Lattice) |
| **ADC** | ADAS3022 (Analog Devices) |
| **DAC** | AD5764R (Analog Devices) |

| I/O Interface | |
|---|---|
| **I/O Connector** | Front I/O 0.8mm Pitch Connector<br>(Honda HDRA-EC100LFDT-SL+)<br>PMC P14 I/O (64 pin Mezzanine Connector)<br>XMC P16 I/O (114 pin Mezzanine Connector) |
| **Digital I/O** | 48 ESD-protected TTL lines<br>TTL signaling voltage level |
| **Number of analog Inputs** | 32 single ended<br>Wherein always two inputs can be combined as one differential. |
| **Analog Input Voltage** | ±20.48V (default)<br>different gains are programmable for each input channel<br>±0.64V up to ±24.576V |
| **Analog Input Filter (typ.)** | 480kHz (-3dB)<br>70kHz (-0.1dB) |
| **Number of analog Output** | 8 single ended |
| **Analog Output Voltage** | ±10V (default) |

<table>
<tr><th colspan="3">Physical Data</th></tr>
<tr><td>Power Requirements</td><td colspan="2">Depends on FPGA design<br>With TXMC635 FPGA Example Design / without external load
<table>
<tr><td></td><td>typical @ +5V VPWR</td><td>typical @ +12V VPWR</td></tr>
<tr><td>TXMC635-xxR</td><td>1.10 A</td><td>0.50 A</td></tr>
</table></td></tr>
<tr><td>Temperature Range</td><td>Operating<br>Storage</td><td>-40°C to +85 °C<br>-40°C to +85°C</td></tr>
<tr><td>MTBF</td><td colspan="2">TXMC635-xxR: 259000 h<br>MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: $G_B$ 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.</td></tr>
<tr><td>Humidity</td><td colspan="2">5 – 95 % non-condensing</td></tr>
<tr><td>Weight</td><td colspan="2">TXMC635-xxR: 124g</td></tr>
</table>

Table 2-1 : Technical Specification

# 3 Handling and Operation Instruction

## 3.1 ESD Protection

**The TXMC635 is sensitive to static electricity. Packing, unpacking and all other handling of the TXMC635 has to be done in an ESD/EOS protected Area.**

## 3.2 Thermal Considerations

**Forced air cooling is recommended during operation. Without forced air cooling, damage to the device can occur.**

## 3.3 Assembling Hints

**When disassembling the TXMC635 from carrier board please keep the mechanical stress as low as possible.**

# 4 PCI Device Topology

The TXMC635 consists of two FPGAs. Both FPGA are designed as a PCIe / PCI endpoint devices. One FPGA is the User FPGA which could be programmed with user defined FPGA code. The second FPGA takes control of on-board hardware functions of TXMC635 and also the configuration control of the User FPGA.

The Configuration FPGA PCI endpoint is connected via a PCI-to-PCIe Bridge to the first x1 Downstream Port of the PCIe Switch (Pericom PI7C9X2G404SL). The User FPGA (Spartan6 PCIe endpoint) is directly connected to the second x1 Downstream Port.

The x1 Upstream Port of the PCIe Switch is connected to the XMC P15 Connector, communicating with the host system.

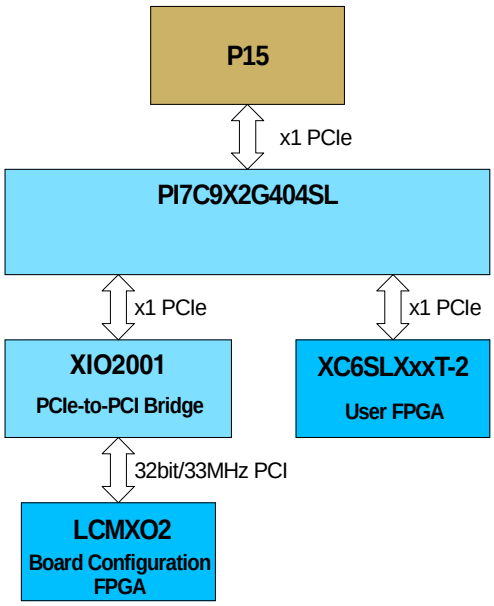



Figure 4-1 : PCIe/PCI Device Topology

| Device | Vendor ID | Device ID | Class Code | Description (as shown by lspci) |
|---|---|---|---|---|
| **PI7C9X2G404SL** | 0x12D8 (Pericom) | 0x2404 | 0x060400 | PCI bridge:<br>0x04h to indicate device as PCI-to-PCI Bridge<br>0x06h to indicate device as Bridge device |
| **XIO2001** | 0x104C (Texas Instruments) | 0x8240 | 0x060400 | PCI bridge: Texas Instruments<br>0x04h to indicate device as PCI-to-PCI Bridge<br>0x06h to indicate device as Bridge device |
| **XC6SLX45T-2 or XC6SLX100T-2** | user defined | | | Device identification for the User programmable FPGA is defined by user. The data will be created with the Spartan-6 PCI Express Endpoint block generation. |
| **LCMXO2** | 0x1498 (TEWS) | 0x927B | 0x068000 | Bridge Device: TEWS Technologies GmbH Device 927B<br>(TXMC635). |

Table 4-1 : On-Board PCIe / PCI Devices

# 4.1 User FPGA (Spartan6)

The User FPGA address map depends on the user application and is not part of this target specification.

# 4.2 Configuration FPGA (MachXO2)

## 4.2.1 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 | 927B 1498 |
| 0x04 | Status | | Command | | Y | 0480 000B |
| 0x08 | Class Code | | | Revision ID | N | 068000 01 |
| 0x0C | BIST | Header Type | PCI Latency Timer | Cache Line Size | Y[7:0] | 00 00 00 08 |
| 0x10 | PCI Base Address 0 for Local Address Space 0 | | | | Y | FFFFFF00 |
| 0x14 | PCI Base Address 1 for Local Address Space 1 | | | | Y | FFFFFF00 |
| 0x18 | PCI Base Address 2 for Local Address Space 2 | | | | N | 00000000 |
| 0x1C | PCI Base Address 3 for Local Address Space 3 | | | | N | 00000000 |
| 0x20 | PCI Base Address 4 for Local Address Space 4 | | | | N | 00000000 |
| 0x24 | PCI Base Address 5 for Local Address Space 5 | | | | N | 00000000 |
| 0x28 | PCI CardBus Information Structure Pointer | | | | N | 00000000 |
| 0x2C | Subsystem ID | | Subsystem Vendor ID | | N | 927B 1498 |
| 0x30 | PCI Base Address for Local Expansion ROM | | | | Y | 00000000 |
| 0x34 | Reserved | | | New Cap. Ptr. | N | 000000 40 |
| 0x38 | Reserved | | | | N | 00000000 |
| 0x3C | Max_Lat | Min_Gnt | Interrupt Pin | Interrupt Line | Y[7:0] | 00 00 01 00 |

Table 4-2 : PCI Configuration Registers

## 4.2.2 PCI BAR Overview

| BAR | Size (Byte) | Space | Prefetch | Port Width (Bit) | Endian Mode | Description |
|---|---|---|---|---|---|---|
| 0 | 256 | MEM | No | 32 | Little | Local Configuration Register Space |
| 1 | 256 | MEM | No | 32 | Little | In-System Programming Data Space |

Table 4-3 : PCI BAR Overview

#### 4.2.2.1 Local Configuration Register Space

| Offset to PCI Base Address | Register Name | Size (Bit) |
|---|---|---|
| 0x00 – 0xBF | Reserved | - |
| 0xC0 | Interrupt Enable Register | 32 |
| 0xC4 | Interrupt Status Register | 32 |
| 0xC8 | Reserved | 32 |
| 0xCC | Reserved | 32 |
| 0xD0 | User FPGA Configuration Control/Status Register | 32 |
| 0xD4 | User FPGA Configuration Data Register (Slave SelectMAP) | 32 |
| 0xD8 | Reserved | 32 |
| 0xDC | Reserved | 32 |
| 0xE0 | ISP Control Register (SPI) | 32 |
| 0xE4 | ISP Configuration Register (SPI) | 32 |
| 0xE8 | ISP Command Register (SPI) | 32 |
| 0xEC | ISP Status Register (SPI) | 32 |
| 0xF0 | Reserved | - |
| 0xF4 | I/O Pull Resistor Configuration Register | 32 |
| 0xF8 | TXMC635 Serial Number | 32 |
| 0xFC | MachXO2 - FPGA Code Version | 32 |

Table 4-4 : Local Configuration Register Space

#### 4.2.2.2 In-System Programming Data Space

The In-System Programming Data Space is used for passing user FPGA configuration data for in-system programming of the User FPGA SPI Flash.

For ISP write/program instructions, the data must be written (zero-based) to the ISP Data Space before the instruction is started. The data must cover a complete SPI Flash memory page.

For ISP read instructions, the data can be read (zero-based) from the ISP Data Space after the instruction is done. The data is passed for a complete SPI Flash memory page.

The ISP Data Space size is 256 byte, covering an SPI Flash Memory Page. All supported SPI Flash read and write instructions are page-based.

Control and status register for In-System programming are located in the Local Configuration Register Space. The data register for direct FPGA in-system programming is also located in the Local Configuration Register Space.

# 5 Register Description

## 5.1 User FPGA

The FPGA register description depends on the user application and is not part of this specification.

## 5.2 Configuration FPGA

### 5.2.1 User FPGA Configuration Control/Status Register - 0xD0

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:6 | | Reserved | | 0 |
| 5 | PULL_CNT | I/O Pull Resistor Controller<br>0: Spartan6 User FPGA controls Pull Resistor<br>1: MachXO2 FPGA controls Pull Resistor | R/W | 0 |
| 4 | S6_LINK_ENA | 1: Spartan6 to PCIe-Switch LINK is enabled<br>0: Spartan6 to PCIe-Switch LINK is disabled | R/W | 1 |
| 3 | FP_INIT_STAT | User FPGA INIT_B Pin Status<br>0: FPGA INIT_B Pin Level is Low (active)<br>1: FPGA INIT_B Pin Level is High (not active) | R | x |
| 2 | FP_DONE_STAT | User FPGA DONE Pin Status<br>The FPGA Done pin is high in case of successful FPGA configuration.<br>0: FPGA DONE Pin Level is Low (not active)<br>1: FPGA DONE Pin Level is High (active) | R | x |
| 1 | FP_RE_CFG | After power-up the FPGA automatically configures from the on-board SPI Flash in 'Master Serial / SPI' mode.<br>User FPGA Re-Configuration<br>1: Set all FPGA I/O pins to High-Z and prepare a User FPGA Re-Configuration<br>1 → 0: Start User FPGA Re-Configuration | R/W | 0 |
| 0 | FP_CFG_MD | Set User FPGA Configuration Mode<br>0: Master Serial / SPI<br>1: Slave SelectMap (Parallel)<br>After power-up the User FPGA automatically configures from the on-board SPI Flash in 'Master Serial / SPI' mode. | R/W | 0 |

Table 5-1 : User FPGA Configuration Control/Status Register

### 5.2.2 User FPGA Configuration Data Register - 0xD4

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | ISP_FP_DAT | ISP Select Map Write Data<br>Write Data Register for direct SelectMap FPGA programming mode<br>Must be written with 32-bit FPGA programming data until the FPGA Done pin goes high (after the actual programming data, writing some dummy data may be required). | w | - |

Table 5-2 : ISP Select Map Data Register

The ISP Select Map Data Register is used to write data within the User FPGA Slave Select Map Configuration directly to the User FPGA.

### 5.2.3 ISP Configuration Register - 0xE4

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:24 | ISP_SPI_ADD | SPI Flash Address A7-A0 | w | 0x00 |
| 23:16 | | SPI Flash Address A15-A8 | w | 0x00 |
| 15:8 | | SPI Flash Address A23-A16 | w | 0x00 |
| 7:0 | ISP_SPI_INS | SPI Flash Instruction Code<br>Supported Instructions:<br>0x02 – Page Program<br>0x20 – Sector Erase<br>0x60 – Chip Erase<br>0x03 – Read Data<br>0x31 – SPI Flash Quad-Mode enable | w | 0x00 |

Table 5-3 : ISP Configuration Register

### 5.2.4 ISP Control Register - 0xE0

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:1 | | Reserved | | 0 |
| 0 | ISP_EN | ISP Mode Enable<br>0: Disable ISP Mode<br>1: Enable ISP Mode<br>This bit controls on-board analog signal multiplexers for signal connections between the MachXO2 CPLD, the User FPGA configuration interface and the on-board SPI Flash.<br>When set, the MachXO2 CPLD is both SPI Flash Master and FPGA Configuration Interface Master.<br>Must be set to 1 for direct SelectMap FPGA or SPI Flash programming.<br>Must be set to 0 when the User FPGA should configure from the SPI Flash (e.g. after SPI Flash programming) in 'Master Serial / SPI' mode.<br>Note, that for ISP Direct FPGA Programming, the FPGA must first be set to 'Slave SelectMap' configuration mode. | R/W | 0 |

Table 5-4: ISP Control Register

### 5.2.5 ISP Command Register - 0xE8

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:2 | | Reserved | - | 0 |
| 1 | ISP_SPI_RST_CMD | ISP SPI Reset Command Bit<br>Writing a '1' sets the Instruction Busy Bit in the ISP Status Register (if not already set).<br>Breaks any ISP SPI instruction in progress and resets the ISP SPI logic.<br>Check the Instruction Busy Bit in the ISP Status Register for reset done status.<br>Always read as '0'. | R/W | 0 |
| 0 | ISP_SPI_INS_CMD | ISP SPI Start Instruction Command Bit<br>Writing a '1' sets the SPI Instruction Busy Bit in the ISP Status Register and starts the configured SPI instruction.<br>Ignored (lost) while the Instruction Busy Bit is set in the ISP Status Register.<br>Always read as '0'. | R/W | 0 |

Table 5-5 : ISP Command Register (SPI)

### 5.2.6 ISP Status Register - 0xEC

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:2 | | Reserved | - | 0x00_0000 |
| 1 | ISP_SPI_ INS_BSY | ISP SPI Instruction Busy Status<br>Set & Cleared automatically by HW.<br>Includes SPI Flash internal program/erase times.<br>When clear again after being set, a new ISP SPI instruction may be started.<br>Capable of generating an event based interrupt.<br>0: No ISP SPI Instruction in Progress<br>1: ISP SPI Instruction in Progress | R | 0 |
| 0 | ISP_SPI_ DAT_BSY | ISP SPI Data Transfer Busy Status<br>Set & Cleared automatically by HW.<br>Does not include SPI Flash internal program/erase times.<br>When clear again after being set, new SPI Flash page data may be written to the ISP Data Space (in program mode) or SPI Flash page data is available in the ISP data space (in read mode).<br>Capable of generating an event based interrupt.<br>0: No ISP SPI Data Transfer in Progress<br>1: ISP SPI Data Transfer in Progress | R | 0 |

Table 5-6 : ISP Status Register

### 5.2.7 Interrupt Enable Register - 0xC0

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:2 | | Reserved | | 0 |
| 1 | ISP_INS_IE | ISP SPI Instruction Done Event Interrupt Enable<br>0: Interrupt Disabled<br>1: Interrupt Enabled<br>While disabled, the corresponding bit in the Interrupt Status Register is ‘0’.<br>Disabling interrupts does not affect the interrupt source. | R/W | 0 |
| 0 | ISP_DAT_IE | ISP SPI Page Data Request Event Interrupt Enable<br>0: Interrupt Disabled<br>1: Interrupt Enabled<br>While disabled, the corresponding bit in the Interrupt Status Register is ‘0’.<br>Disabling interrupts does not affect the interrupt source. | R/W | 0 |

Table 5-7 : Interrupt Enable Register

### 5.2.8 Interrupt Status Register - 0xC4

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:2 | | Reserved | | 0 |
| 1 | ISP_INS_IS | ISP SPI Instruction Done Event Interrupt Status<br>When set, the PCI INTA# interrupt is asserted.<br>The Interrupt is cleared by writing a '1'.<br>0: Interrupt not active or disabled<br>1: Interrupt active and enabled | R/C | 0 |
| 0 | ISP_DAT_IS | ISP SPI Page Data Done Event Interrupt Status<br>When set, the PCI INTA# interrupt is asserted.<br>The Interrupt is cleared by writing a '1'.<br>0: Interrupt not active or disabled<br>1: Interrupt active and enabled | R/C | 0 |

Table 5-8 : Interrupt Status Register

### 5.2.9 I/O Pull Resistor Configuration Register - 0xF4

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:8 | | Reserved | - | 0 |
| 7:6 | PULL_G3 | I/O Group pull-up / pull-down selecton<br>Value could be changed only if PULL_CNT is set to MachXO2 controlling.<br>11 : pull-down<br>10 : pull-up to 3.3V<br>01 : pull-up to 5V<br>00 : No pull-up or pull-down<br><br>I/O lines are summarized in the following groups.<br>G0 = I/O_0 … I/O_15<br>G1 = I/O_16 … I/O_31<br>G2 = I/O_32 … I/O_47<br>G3 = not used on TXMC635 | R/W | User FPGA Pin adjustment |
| 5:4 | PULL_G2 | | | |
| 3:2 | PULL_G1 | | | |
| 1:0 | PULL_G0 | | | |

Table 5-9 : I/O Pull-Resistor Configuration Register

Each TTL I/O Line has a 4k7 Pull-Resistor. The 48 I/O Lines are divided into three groups which can be configured as 3.3V pull-up, 5V pull-up or pull-down. In addition, the Pull-Resistors can float.

If the Pull-Resistors float, the user should keep in mind that the 16 I/O Lines of the group are connected via their Pull-Resistors.

**The default adjustment is that the USER FPGA code must control the I/O Pull Configuration depending on USER FPGA I/O Function (see also chapter “I/O Pull Configuration”).**

### 5.2.10 TXMC635 Serial Number - 0xF8

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | S_NUMBER | The value is the unique serial number of each TXMC635 module | R | - |

Table 5-10: TXMC635 Serial Number

Example: 0x008F_DD0F => SNo.: 9428239

**The serial number can also be read via an I2C interface from Spartan6.**

### 5.2.11 MachXO2 - FPGA Code Version - 0xFC

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | CODE_VER | The value shows the MachXO2 FPGA Firmware code version of the TXMC635 module. | R | - |

Table 5-11: MachXO2 - FPGA Code Version

Example:

0x0000_0100 => bit 32 downto 24 : reserved

0x0000_0100 => bit 23 downto 16 : Major FPGA Code Version

0x0000_0100 => bit 23 downto 16 : Minor FPGA Code Version

# 6 Interrupts

## 6.1 Interrupt Sources

### 6.1.1 User FPGA (Spartan6)

The FPGA interrupt sources depend on the user application and are not part of this target specification.

### 6.1.2 Configuration FPGA (MachXO2)

The Configuration FPGA provides two interrupt sources. Both interrupts are only available during SPI programming instructions. The Slave Select Map Mode does not provide interrupt support.

- ISP SPI Instruction Done Event Interrupt

  Event-based interrupt that becomes active, when the ISP SPI Instruction Busy status bit changes from busy to not-busy.

- ISP SPI Page Data Done Event Interrupt

  Event-based interrupt that becomes active, when the ISP SPI Data Busy status bit changes from busy to not-busy.

## 6.2 Interrupt Handling

### 6.2.1 User FPGA (Spartan6)

The interrupt handling depends on the user application and is not part of this target specification.

### 6.2.2 Configuration FPGA (MachXO2)

Both Interrupts of the MachXO2 FPGA must be cleared via writing access to the corresponding Interrupt Status Flag in the Interrupt Status Register.

# 7 Functional Description

## 7.1 User FPGA Block Diagram

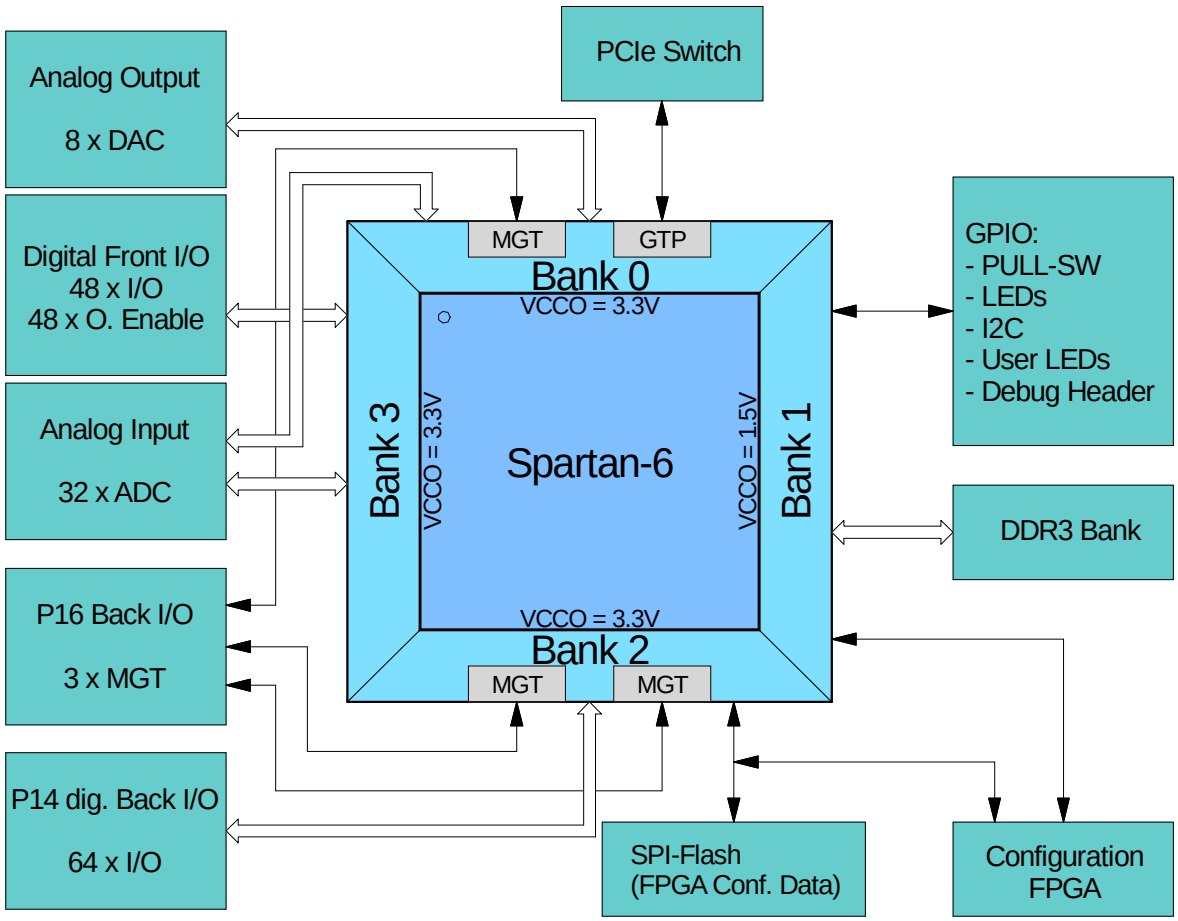



Figure 7-1 : FPGA Block Diagram

## 7.2 User FPGA

The FPGA is a Spartan-6 LX45T-2 or LX100T-2 FPGA. Each Spartan-6 FPGA in a FGG484 package provides two Memory Controller Blocks and one Endpoint Block for PCI Express (x1 Linkage).

| Spartan-6 | Slices | Flip-Flops | DSP48A1 Slices | Block RAM (Kb) | CMTs | GTP Transceivers |
|---|---|---|---|---|---|---|
| LX45T | 6.822 | 54.576 | 58 | 2.088 | 4 | 4 |
| LX100T | 15.822 | 126.576 | 180 | 4.824 | 6 | 4 |

Table 7-1 : TXMC635 FPGA Feature Overview

The board supports JTAG, master serial mode configuration from SPI-Flash or SelectMAP configuration via Configuration FPGA Register.

The FPGA is equipped with 4 I/O banks and 4 MGT (multi gigabit transceiver).

| Bank | VCCO | VREF | Signals | Note |
|---|---|---|---|---|
| Bank 0 | 3.3V | none | dig. Front I/O Interface<br>ser. ADC Interface<br>ser. DAC Interface | |
| Bank 1 | 1.5V | 0.75V | DDR3 Bank<br>Local Bus Interface | +GPIO / LED /Debug |
| Bank 2 | 3.3V | none | dig. Back I/O Interface | +Configuration |
| Bank 3 | 3.3V | none | dig. Front I/O Interface<br>ser. ADC Interface<br>ser. DAC Interface | |
| GTP0 | PCIe Interface to PCIe Switch Device | | | |
| GTP1 | optional used for Back I/O link | | | |
| GTP2 | optional used for Back I/O link | | | |
| GTP3 | optional used for Back I/O link | | | |

Table 7-2 : FPGA Bank Usage

The FPGA's VCCAUX is connected to the 3.3V supply.

# 7.3 User FPGA Gigabit Transceiver (GTP)

The TXMC635 provides one MGT as Spartan-6 PCI Express Endpoint Block and three MGT for high speed XMC P16 interface.

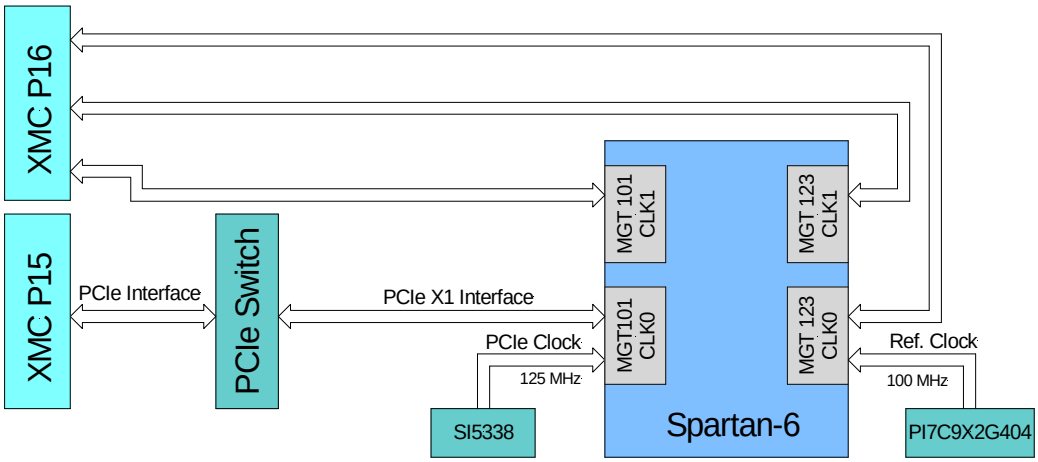



Figure 7-2 : GTP Block Diagram

| GTP | Signal | FPGA Pins | Connected to |
|---|---|---|---|
| MGT0_101 | MGTTX | B6 / A6 | used for PCI Express Endpoint Block |
| | MGTRX | D7 / C7 | |
| MGT1_101 | MGTTX | B8 / A8 | connected to XMC P16 |
| | MGTRX | D9 / C9 | |
| MGT0_123 | MGTTX | B14 / A14 | connected to XMC P16 |
| | MGTRX | D13 / C13 | |
| MGT1_123 | MGTTX | B16 / A16 | connected to XMC P16 |
| | MGTRX | D15 / C15 | |

Table 7-3 : MGT Connections

The MGT clock MGT0_101 (PCI Express Endpoint Block clock reference) of 125 MHz is generated by the SI5338 low-jitter clock generator. The MGT0_123 is connected directly to the PCIe-Switch (PI7C9X2G404) reference clock. MGT1_101 and MGT1_123 are not used on the TXMC635.

| GTP | Signal | FPGA Pins | Connected to |
|---|---|---|---|
| MGT0_101 | MGTREFCLK | A10 / B10 | 125 MHz (derived SI5338 clock generator) |
| MGT1_101 | MGTREFCLK | C11 / D11 | not connected |
| MGT0_123 | MGTREFCLK | A12 / B12 | 100 MHz from PCIe-Switch |
| MGT1_123 | MGTREFCLK | E12 / F12 | not connected |

Table 7-4 : Multi Gigabit Transceiver Reference Clocks

## 7.4 User FPGA Configuration

The Spartan6 could be configured by the following interfaces:

- Master Serial SPI Flash Configuration Interface
- JTAG Interface via JTAG Header or TEWS Debug connector
- PCIe Interface via MachXO2 Configuration FPGA Slave Select Map Interface Configuration

The change of the configuration mode is done with a configuration register of the MachXO2 FPGA.

**At Power-up, the TXMC635 Spartan-6 FPGA always configures via x4 SPI Interface by "Master Serial / SPI" mode.**

**On delivery the SPI Flash contains the TEWS example application for the TXMC635 Spartan6 device.**

### 7.4.1 Master Serial SPI Flash Configuration

It is important for User FPGA Configuration via SPI Master Mode that the ISP Mode Enable (ISP_EN) is to disable the ISP Mode. This is the default value after Power Up.
See also Register Description of TXMC635 Configuration Device.

## 7.4.2 Manually User FPGA SPI Flash Reconfiguration

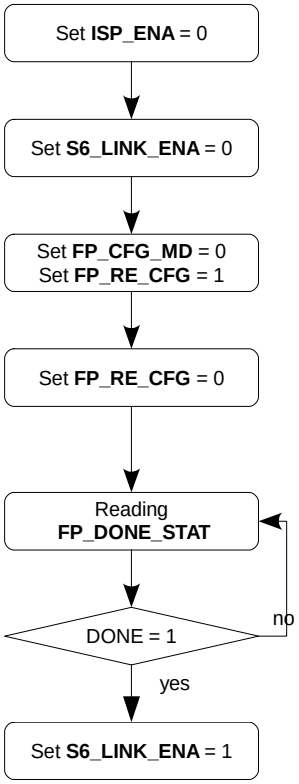
A manually User FPGA Reconfiguration could be performed with the User FPGA Reconfigure Command in the Global Configuration Register.

Set the User FPGA Reconfigure Command to set the User FPGA to configuration state with all FPGA I/O pins are High-Z.

Use the following procedure to perform a User FPGA SPI Re-configuration

- Assure that ISP Mode Enable is disabled.
- By Re-configuring the Spartan6 the XILINX PCIe endpoint is reloaded and is temporarily not available on the PCI bus. To avoid error messages of the PCIe switch the link between the PCIe Switch and the Spartan6 is disabled.
- Set the User FPGA Configuration Mode (FP_CFG_MD) to Master Serial / SPI and prepare the FPGA Re-Configuration.
- Start the FPGA Re-configuration by setting the FP_RE_CFG bit of the User FPGA Configuration Control/Status Register to 0.
- Assure that the FPGA DONE Pin status shows a successful FPGA Configuration.

  0: FPGA DONE Pin Level is Low (FPGA is not configured)
  1: FPGA DONE Pin Level is High (FPGA is configured)
- The link between the PCIe Switch and the Spartan6 must be enabled.




A successful User FPGA configuration is indicated with FPGA_DONE status in the Global Status Register and the on-board User FPGA Done LED.

It must be considered in any case, that the Re-configuration of the User FPGA also Re-configures the PCIe Endpoint of the User FPGA. This has the consequence that the PCI Header of the User FPGA PCIe Endpoint no longer exists. For this purpose it is necessary to disable the link between the PCIe switch and the User FPGA PCIe Endpoint before preparing the FPGA Re-configuration and to enable the link again after Re-configuration.

Additionally, after FPGA Re-Configuration the User FPGA PCIe Endpoint PCI Header must be configured again. If the PCIe interface of the User FPGA PCIe Endpoint does not change. Device ID, Vendor ID, Class Code and PCI bars do not change, the PCI header could be saved before the FPGA Re-configuration and written back to configuration space after the Re-configuration.

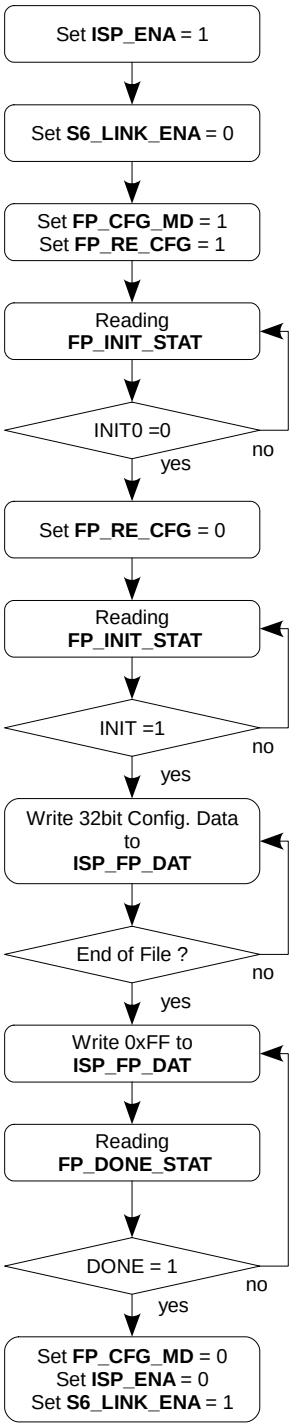
### 7.4.3 Slave Select Map Configuration

For direct User FPGA configuration via PCIe Interface the **User FPGA Configuration Mode** must be set to **Slave SelectMap** Mode. The on-board logic sets the User FPGA in configuration state with all FPGA I/O pins switches to High-Z. User FPGA is now ready for new configuration data.

The following procedure is required for Select Map Mode User FPGA configuration / Re-configuration.

- First the In System Program (ISP) Mode must be enabled.
- By Re-configuring the Spartan6 the XILINX PCIe endpoint is reloaded and is temporarily not available on the PCI bus. To avoid error messages of the PCIe switch the link between the PCIe Switch and the Spartan6 is disabled.
- Check response of the Spartan6 by reading the FPGA INIT_B pin value. If the Level is low the Spartan6 FPGA is in Reset Mode, and then configuration process could be continued.
- Start the FPGA Re-configuration by setting the FP_RE_CFG bit of the User FPGA Configuration Control/Status Register to 0.
- Check response of the Spartan6 by reading the FPGA INIT_B pin value. While the FPGA INIT_B pin Level is low the Spartan6 isn't ready for configuration.
- If FPGA INIT_B pin high then the configuration data must be continually written to the ISP SelectMap Data Register. Typically 373103 PCI write accesses are required to configure a Spartan6 6SLX45T.
- Dummy Write accesses to create configuration clock cycles while FP_DONE_STAT is low.
- A successful configuration of the User FPGA is indicated with FP_DONE_STAT in the User FPGA Configuration Control/Status Register and the on-board User FPGA Done LED.

  0: FPGA DONE Pin Level is Low (FPGA is not configured)
  1: FPGA DONE Pin Level is High (FPGA is configured)
- After Re-configuration was successful the User FPGA Configuration Mode and the ISP Mode could be disabled. Also the link between the PCIe Switch and the Spartan6 must be enabled.



**If not all configuration data bytes are written the User FPGA is not configured correctly.**

The number of bytes that must be written corresponds to the size of the XILINX configurations files. Typically the .bin or the .bit file could be used as data source.

The .bit file is the standard generated programming file. This is a binary configuration data file which contains header information that does not need to be downloaded to the FPGA. For generating the .bin file the BitGen option **–g Binary:yes** must be used. This is also a binary configuration data file but without header information. For configure the Spartan6 FPGA of the TXMC635 both files could be used. Both binary configuration data file have addition data to the actual configuration data.

Two examples are provided here. In the .bit file the data can be used from the offset 0x000000b0. For the .bin file the data can be used starting at offset 0x00000050.

**Example .bit file:**

```
0x00000000 00 09 0f f0 0f f0 0f f0 0f f0 00 00 01 61 00 2c bit-file
0x00000010 78 69 6c 69 6e 78 5f 70 63 69 65 5f 31 5f 31 5f header
0x00000020 65 70 5f 73 36 2e 6e 63 64 3b 55 73 65 72 49 44
0x00000030 3d 30 78 46 46 46 46 46 46 46 46 00 62 00 0e 36
0x00000040 73 6c 78 34 35 74 66 67 67 34 38 34 00 63 00 0b
0x00000050 32 30 31 34 2f 30 34 2f 32 38 00 64 00 09 31 34
0x00000060 3a 34 37 3a 31 34 00 65 00 16 a6 b8 ff ff ff ff add. S6
0x00000070 ff ff ff ff ff ff ff ff ff ff ff ff aa 99 55 66 Config.
0x00000080 31 e1 ff ff 32 61 00 44 32 81 6b 00 32 a1 00 44 Data
0x00000090 32 c1 6b 00 32 e1 00 00 30 a1 00 00 33 01 31 00
0x000000a0 32 01 00 5f 30 a1 00 0e 20 00 20 00 20 00 20 00
0x000000b0 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff TXMC635
0x000000c0 aa 99 55 66 30 a1 00 07 20 00 31 a1 06 28 31 41 Config.
0x000000d0 3d 00 31 61 09 ee 31 c2 04 02 80 93 30 e1 00 cf Data
0x000000e0 30 c1 00 81 20 00 20 00 20 00 20 00 20 00 20 00
0x000000f0 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
...
0x0016aa60 20 00 20 00 20 00 20 00 20 00
```

**Example .bin file:**

```
0x00000000 ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff add. S6
0x00000010 aa 99 55 66 31 e1 ff ff 32 61 00 44 32 81 6b 00 Config.
0x00000020 32 a1 00 44 32 c1 6b 00 32 e1 00 00 30 a1 00 00 Data
0x00000030 33 01 31 00 32 01 00 5f 30 a1 00 0e 20 00 20 00
0x00000040 20 00 20 00 ff ff ff ff ff ff ff ff ff ff ff ff TXMC635
0x00000050 ff ff ff ff aa 99 55 66 30 a1 00 07 20 00 31 a1 Config.
0x00000060 06 28 31 41 3d 00 31 61 09 ee 31 c2 04 02 80 93 Data
0x00000070 30 e1 00 cf 30 c1 00 81 20 00 20 00 20 00 20 00
0x00000080 20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
...
0x0016aa60 20 00 20 00 20 00 20 00 20 00
```

See also the XILINX User Guide (ug380) “Spartan6 FPGA Configuration” for more information about Configuration Details and Configuration Data File Formats.

## 7.4.4 Configuration via JTAG

The TXMC635 provides two JTAG chains which are accessible by one of the following connector options:

User JTAG Chain

- 14-pin JTAG Header
- Debug Connector

TEWS Factory configuration Chain

- XMC Connector P15

The User JTAG Chain is accessible from the JTAG Header or from the Debug Connector. These interfaces are connected in parallel, so only one connection should be made to avoid signal contentions/possible hardware damage.

For direct FPGA configuration, FPGA read back or in-system diagnostics with ChipScope, the JTAG Header can be used to access the JTAG-chain. Also an indirect SPI – PROM programming is possible via JTAG Chain.

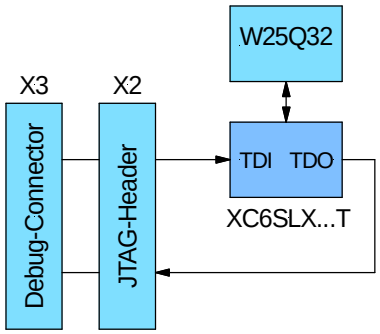



Figure 7-3 : User JTAG-Chain

The TEWS Factory JTAG Chain is accessible from the XMC P15 connector.

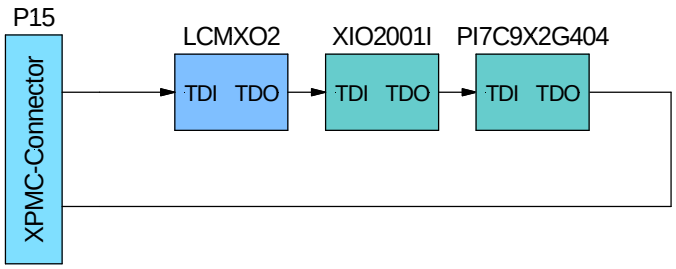


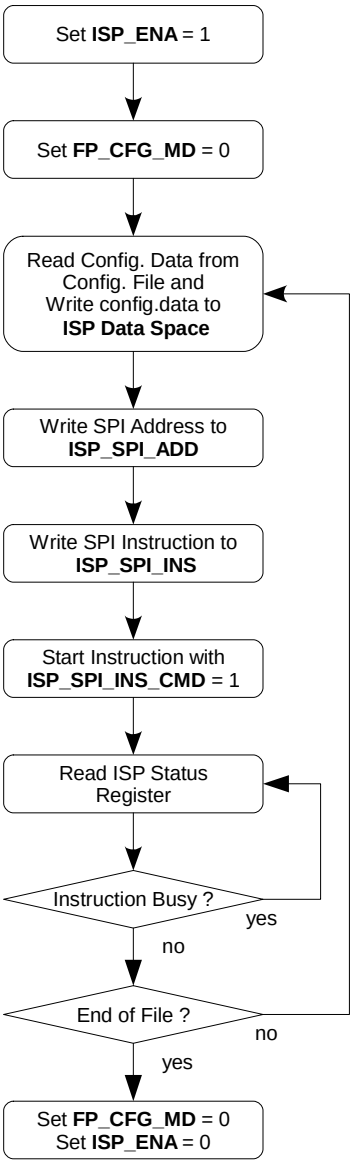

Figure 7-4 : TEWS Factory JTAG-Chain

## 7.4.5 Programming User FPGA SPI Configuration Flash

For programming the User FPGA SPI Configuration Flash the **User FPGA Configuration Mode** must be set to **Master Serial / SPI** and the ISP Mode must be enabled.

The following procedure is required for User FPGA SPI Configuration Flash programming and subsequent reconfiguration of the User FPGA.

- Enable then ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash.
- Read Configuration data from Configuration File and write Data to the In Circuit Programming Data Space. 256Byte (1 SPI Flash page) each time can be programmed maximum.
- Set the programming start address and write instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for next write instruction.
- Process should be repeated until all configuration data is written to the SPI Flash
- After completion the data programming, the ISP Mode bit must cleared to set configuration path to User FPGA and a Reconfiguration could be performed.




A successful configuration of the User FPGA is indicated with FP_DONE_STAT in the User FPGA Configuration Control/Status Register and the on-board User FPGA Done LED.

**The Programming Instruction always starts at address 0x00 to write data from the ISP Programming Data Space to the SPI flash.**

**If not all configuration data bytes are written the User FPGA is not configured correctly.**

The source for the User FPGA SPI Configuration Flash data should be the .mcs file. This file format can be created from the .bit file by using the XILINX iMPACT or PROMGen software. Besides the pure configuration data the .mcs file format includes SPI Flash specific configuration data. These data are needed to ensure a correct configuration of the User FPGA from the SPI PROM.

For how to generate the .mcs file out of the .bit file by using XILINX iMPACT or PROMGen software please refer to the XILINX iMPACT documentation.

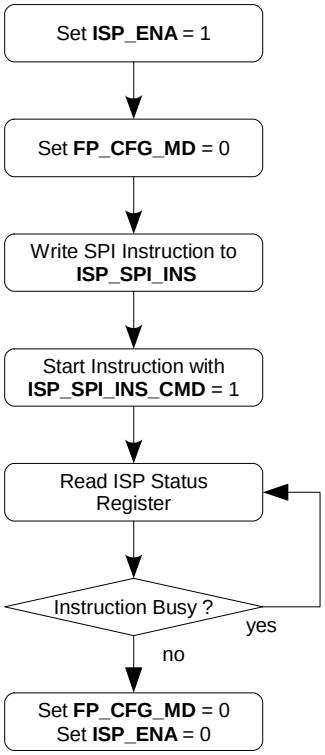
**TXMC635-xxR User FPGA SPI Configuration device:**

SPI Flash: Winbond W25Q32FV or W25Q32JV; 32M; Data Width = 4 bit

## 7.4.6 Erasing User FPGA SPI Configuration Flash

For Chip Erasing the User FPGA SPI Configuration Flash the **User FPGA Configuration Mode** must be set to **Master Serial / SPI** and the ISP Mode must be enabled.

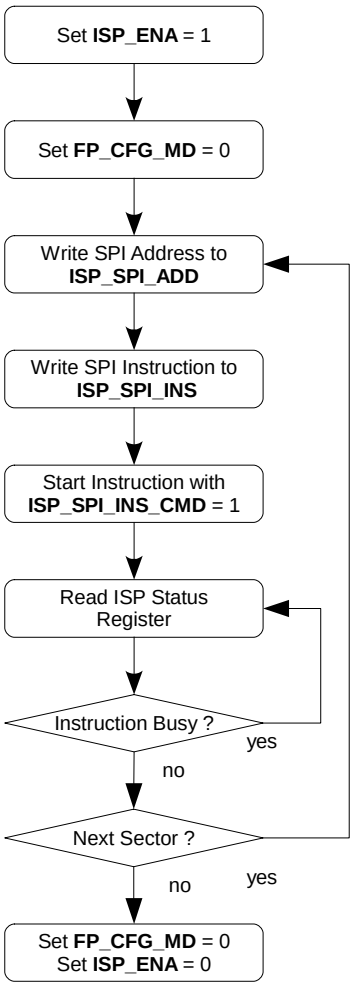
- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that the User FPGA Configuration Mode is set to SPI Flash.
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for erasing process end.
- After completion of the erasing process, the ISP Mode bit should be cleared to set configuration path to User FPGA or a User FPGA SPI Configuration Flash programming process could be done.



## 7.4.7 Sector Erasing User FPGA SPI Configuration Flash

For Sector Erasing the User FPGA SPI Configuration Flash the **User FPGA Configuration Mode** must be set to **Master Serial / SPI** and the ISP Mode must be enabled.

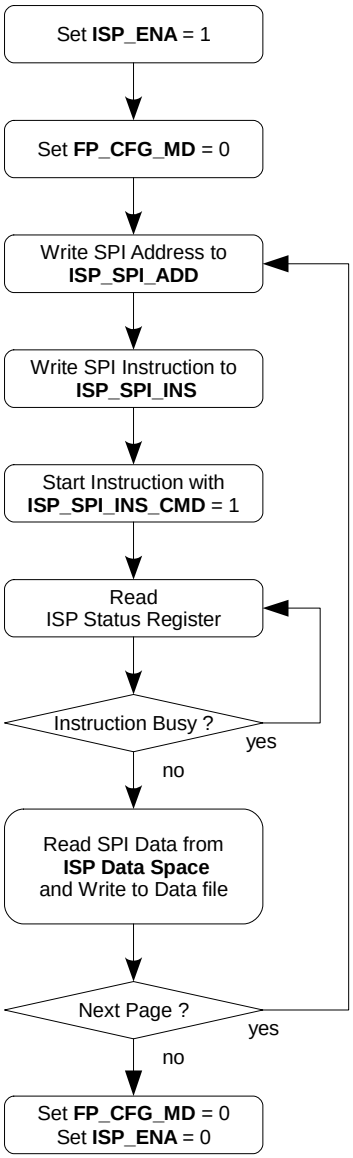
- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that the User FPGA Configuration Mode is set to SPI Flash.
- Write the Sector Address to the ISP Configuration Register
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for erasing process end.
- Process could be repeated for other sectors.
- After completion of the erasing process, the ISP Mode bit should be cleared to set configuration path to User FPGA or a User FPGA SPI Configuration Flash programming process could be done.



## 7.4.8 Reading User FPGA SPI Configuration Flash

For Reading the User FPGA SPI Configuration Flash the **User FPGA Configuration Mode** must be set to **Master Serial / SPI** and the ISP Mode must be enabled.

- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that the User FPGA Configuration Mode is set to SPI Flash.
- Set the reading start address and write instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for next write instruction.
- Read one page of SPI Data from In Circuit Programming Data Space and write to Data file
- Process could be repeated until all needed data are written to the Data file.
- After completion of the reading process, the ISP Mode bit must cleared to set configuration path back to User FPGA.



## 7.4.9 Generate Spartan6 Configuration Data

The TXMC635 must be configured to use the 32 MHz external master clock as CCLK.

The following Xilinx BitGen options must be used or set in the Xilinx ISE design properties:

| | |
|---|---|
| 'Enable External Master Clock' (-g ExtMasterCclk_en) | = enable |
| 'Setup External Master Clock Devision' (-g ExtMasterCclk_divide) | = 1 |

To use the maximum data transfer speed of the User FPGA SPI Configuration Flash the SPI Configuration Bus Width must be set to the x4.

| | |
|---|---|
| 'Set SPI Configuration Bus Width' (-g SPI_buswidth) | = 4 |

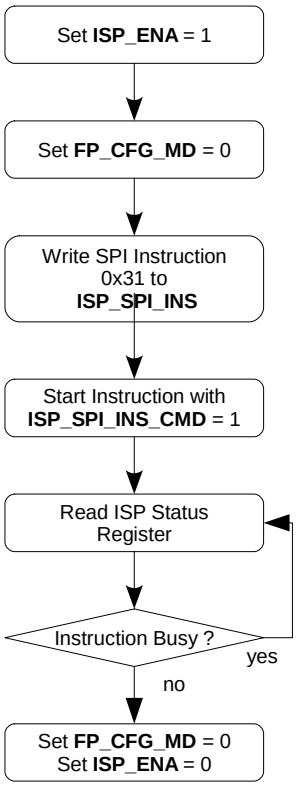
Without this option, the configuration time for the Spartan6 FPGA exceeds the maximum PCIe bus setup time.

## 7.4.10 SPI-PROM Quad Mode enable

Be due to the required SPI Configuration Bus Width X4 the Quad-Mode of the SPI Flash must always be enabled. Therefore the SPI Flash on the TXMC635 provides a non-volatile bit. This bit is set during TEWS factory test and programming process. The Quad-Mode enable bit is non-volatile, so it is not necessary to re-enable this bit every SPI Flash programming process. Even when programming the SPI Flash with the Xilinx iMPACT tool this bit is automatically programmed. If this bit is not set the Spartan6 FPGA could not be configured from the SPI Flash.

If this bit should ever be cleared, it can be reprogrammed using the SPI Flash Quad-Mode enable instruction.

- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that the User FPGA Configuration Mode is set to SPI Flash.
- Set the SPI Flash Quad-Mode enable instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done for successful process end.
- After completion of the instruction process, the ISP Mode bit must be cleared to set configuration path back to User FPGA.



### 7.4.11 Board Configuration FPGA

The Board Configuration FPGA (BCF) is factory configured, and handles the basic board setup.

**Changing or erase the BCF content leads to an inoperable TXMC635 FPGA configuration.**

# 7.5 Clocking

## 7.5.1 FPGA Clock Sources

As a central clock generator of TXMC635 the Si5338 clock generator is used. This provides all necessary clocks for the User FPGA and the Configuration FPGA.

The following figure depicts an abstract User FPGA clock flow.

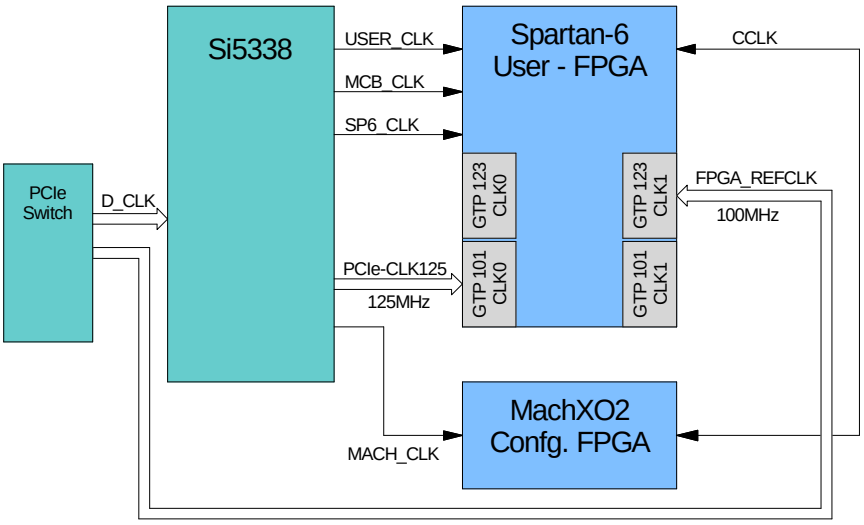



Figure 7-5 : FPGA Clock Sources

The following table lists the available clock sources on the TXMC635:

| FPGA Clock-Pin Name | FPGA Pin Number | Source | Description |
|---|---|---|---|
| MGTREFCLK0_101 | A10 / B10 | SI5338 low-jitter clock generator | 125 MHz<br>PCIe Reference clock |
| MGTREFCLK0_113 | A12 / B12 | PCIe Switch PI7C9XG404 | 100 MHz differential Reference clock input |
| IO_L30P_GCLK1_2 | Y13 | SI5338 low-jitter clock generator | MCB CLK<br>62.5 MHz |
| IO_L43P_GCLK22_3 | M5 | SI5338 low-jitter clock generator | USER CLK<br>83.3325 MHz |
| IO_L30N_GCLK0_2 | AB13 | SI5338 low-jitter clock generator | 32.00 MHz Clock Input<br>After configuration this clock could be used by FPGA design. |
| IO_L30N_GCLK0_USERCCLK_2 | Y20 | Configuration FPGA | Used for external configuration clock (CCLK) |

Table 7-5 : Available FPGA clocks

**Note:**

**Since the PCIe clock of the PCIe switch is used as the source for the clock generator, all 4 clock generated by the SI5338 clock generator also depend on it.**

**This means that if the PCIe host system uses the 'spread-spectrum' feature for the PCIe clock, all outputs of the SI5338 are also affected.**

**This may cause problems or inaccuracies with equidistant functions in the FPGA. The solution would**

# 7.6 Front I/O Interface

Each of the 48 digital front IO channels is realized with single ended or differential digital buffers. Each channel provides an I/O data signal and an output enable signal which is direct connected to the FPGA device.

The I/O channels are accessible through the I/O Bank 0 and Bank 3 of the Spartan-6 FPGA. The subsequent table lists required I/O setting for correct interfacing.

| Signal Name | Pin Number | Direction | IO Standard | IO Bank | Drive [mA] | Slew Rate |
|---|---|---|---|---|---|---|
| FPGA_IO<0># | P1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<1># | P6 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<2># | AA2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<3># | W1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<4># | AA1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<5># | V1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<6># | W3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<7># | W4 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<8># | T1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<9># | P2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<10># | F2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<11># | H2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<12># | K1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<13># | J1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<14># | F1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<15># | B1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<16># | T2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<17># | R3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<18># | T5 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<19># | V5 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<20># | L4 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<21># | M3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<22># | M4 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<23># | A3 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<24># | C1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<25># | B2 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<26># | D17 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<27># | D18 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<28># | K2 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<29># | D1 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<30># | A2 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<31># | F3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |

| | | | | | | |
|---|---|---|---|---|---|---|
| FPGA_IO<32># | E3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<33># | J3 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<34># | E4 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<35># | D5 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<36># | A4 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<37># | A5 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<38># | F5 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<39># | H6 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<40># | C5 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<41># | K5 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<42># | F8 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<43># | L3 | IN/OUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_IO<44># | H5 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<45># | G6 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<46># | G7 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_IO<47># | K7 | IN/OUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<0># | N1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<1># | M1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<2># | Y1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<3># | V2 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<4># | Y2 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<5># | Y3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<6># | U1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<7># | V3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<8># | U3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<9># | N3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<10># | N6 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<11># | P8 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<12># | R7 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<13># | M2 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<14># | G1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<15># | D3 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<16># | R1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<17># | T3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<18># | P5 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<19># | U4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<20># | H4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<21># | J7 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<22># | P4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<23># | T6 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<24># | N7 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |

| | | | | | | |
|---|---|---|---|---|---|---|
| FPGA_OE<25># | R4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<26># | G13 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<27># | F17 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<28># | H1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<29># | E1 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<30># | D2 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<31># | B3 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<32># | M6 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<33># | H3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<34># | C4 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<35># | G4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<36># | G3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<37># | K3 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<38># | J4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<39># | T4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<40># | D4 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<41># | K4 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<42># | F7 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |
| FPGA_OE<43># | F9 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<44># | E5 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<45># | E6 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<46># | J6 | OUTPUT | LVCMOS33 | 0 | 8 | SLOW |
| FPGA_OE<47># | G8 | OUTPUT | LVCMOS33 | 3 | 8 | SLOW |

Table 7-6 : Digital Front I/O Interface

## 7.7 Back I/O Interface

P14 Back I/O Pins of the TXMC635 are direct routed to the Spartan6 FPGA. The I/O functions of these FPGA pins are directly dependent on the configuration of the FPGA.

The Spartan6 VCCO voltage is set to 3.3V, so only the 3.3V I/O standards LVCMOS33, LVTTL33 and LVDS_33 are possible for using on TXMC635 back I/O interface.

| Signal Name | Pin Number | Direction | IO Standard for example | IO Bank |
|---|---|---|---|---|
| BACK_IO0+ | AA4 | IN/OUT | LVDS_33 | 2 |
| BACK_IO0- | AB4 | IN/OUT | LVDS_33 | 2 |
| BACK_IO1+ | W6 | IN/OUT | LVDS_33 | 2 |
| BACK_IO1- | Y6 | IN/OUT | LVDS_33 | 2 |
| BACK_IO2+ | T7 | IN/OUT | LVDS_33 | 2 |
| BACK_IO2- | U6 | IN/OUT | LVDS_33 | 2 |
| BACK_IO3+ | Y7 | IN/OUT | LVDS_33 | 2 |
| BACK_IO3- | AB7 | IN/OUT | LVDS_33 | 2 |
| BACK_IO4+ | V7 | IN/OUT | LVDS_33 | 2 |
| BACK_IO4- | W8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO5+ | AA8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO5- | AB8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO6+ | T8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO6- | U8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO7+ | R9 | IN/OUT | LVDS_33 | 2 |
| BACK_IO7- | R8 | IN/OUT | LVDS_33 | 2 |
| BACK_IO8+ | Y9 | IN/OUT | LVDS_33 | 2 |
| BACK_IO8- | AB9 | IN/OUT | LVDS_33 | 2 |
| BACK_IO9+ | U9 | IN/OUT | LVDS_33 | 2 |
| BACK_IO9- | V9 | IN/OUT | LVDS_33 | 2 |
| BACK_IO10+ | AA10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO10- | AB10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO11+ | W10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO11- | Y10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO12+ | T10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO12- | U10 | IN/OUT | LVDS_33 | 2 |
| BACK_IO13+ | Y11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO13- | AB11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO14+ | V11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO14- | W11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO15+ | R11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO15- | T11 | IN/OUT | LVDS_33 | 2 |
| BACK_IO16+ | AA12 | IN/OUT | LVDS_33 | 2 |

| | | | | |
|---|---|---|---|---|
| BACK_IO16- | AB12 | IN/OUT | LVDS_33 | 2 |
| BACK_IO17+ | W12 | IN/OUT | LVDS_33 | 2 |
| BACK_IO17- | Y12 | IN/OUT | LVDS_33 | 2 |
| BACK_IO18+ | T12 | IN/OUT | LVDS_33 | 2 |
| BACK_IO18- | U12 | IN/OUT | LVDS_33 | 2 |
| BACK_IO19+ | V13 | IN/OUT | LVDS_33 | 2 |
| BACK_IO19- | W13 | IN/OUT | LVDS_33 | 2 |
| BACK_IO20+ | AA14 | IN/OUT | LVDS_33 | 2 |
| BACK_IO20- | AB14 | IN/OUT | LVDS_33 | 2 |
| BACK_IO21+ | W14 | IN/OUT | LVDS_33 | 2 |
| BACK_IO21- | Y14 | IN/OUT | LVDS_33 | 2 |
| BACK_IO22+ | U14 | IN/OUT | LVDS_33 | 2 |
| BACK_IO22- | U13 | IN/OUT | LVDS_33 | 2 |
| BACK_IO23+ | Y15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO23- | AB15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO24+ | AA16 | IN/OUT | LVDS_33 | 2 |
| BACK_IO24- | AB16 | IN/OUT | LVDS_33 | 2 |
| BACK_IO25+ | Y16 | IN/OUT | LVDS_33 | 2 |
| BACK_IO25- | W15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO26+ | U16 | IN/OUT | LVDS_33 | 2 |
| BACK_IO26- | V15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO27+ | T15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO27- | U15 | IN/OUT | LVDS_33 | 2 |
| BACK_IO28+ | Y17 | IN/OUT | LVDS_33 | 2 |
| BACK_IO28- | AB17 | IN/OUT | LVDS_33 | 2 |
| BACK_IO29+ | AA18 | IN/OUT | LVDS_33 | 2 |
| BACK_IO29- | AB18 | IN/OUT | LVDS_33 | 2 |
| BACK_IO30+ | W17 | IN/OUT | LVDS_33 | 2 |
| BACK_IO30- | Y18 | IN/OUT | LVDS_33 | 2 |
| BACK_IO31+ | V17 | IN/OUT | LVDS_33 | 2 |
| BACK_IO31- | W18 | IN/OUT | LVDS_33 | 2 |

Table 7-7 : Digital Back I/O Interface

# 7.8 Memory

The TXMC635 is equipped with a 128 Mbytes, 16 bit wide DDR3 SDRAM and a 32-Mbit non-volatile SPI-Flash. The SPI-Flash can also be used as configuration memory.

## 7.8.1 DDR3 SDRAM

The TXMC635 provides a MT41… (96-ball) DDR3 memory device. The memory is accessible through the Memory Controller Block hard-IPs in bank 1 of the Spartan-6 FPGA.

The memory component's CS# is fixed to GND. The address bits A14 and A13 are memory address expansion bits.

| Signal | DDR Bank A FPGA Pin | I/O Standard | Termination | Memory Device | |
|---|---|---|---|---|---|
| | | | | Pin | Name |
| A0 | H21 | SSTL15_II | 49.9Ω $V_{TT}$ | N3 | A0 |
| A1 | H22 | SSTL15_II | 49.9Ω $V_{TT}$ | P7 | A1 |
| A2 | G22 | SSTL15_II | 49.9Ω $V_{TT}$ | P3 | A2 |
| A3 | J20 | SSTL15_II | 49.9Ω $V_{TT}$ | N2 | A3 |
| A4 | H20 | SSTL15_II | 49.9Ω $V_{TT}$ | P8 | A4 |
| A5 | M20 | SSTL15_II | 49.9Ω $V_{TT}$ | P2 | A5 |
| A6 | M19 | SSTL15_II | 49.9Ω $V_{TT}$ | R8 | A6 |
| A7 | G20 | SSTL15_II | 49.9Ω $V_{TT}$ | R2 | A7 |
| A8 | E20 | SSTL15_II | 49.9Ω $V_{TT}$ | T8 | A8 |
| A9 | E22 | SSTL15_II | 49.9Ω $V_{TT}$ | R3 | A9 |
| A10 | J19 | SSTL15_II | 49.9Ω $V_{TT}$ | L7 | A10/AP |
| A11 | H19 | SSTL15_II | 49.9Ω $V_{TT}$ | R7 | A11 |
| A12 | F22 | SSTL15_II | 49.9Ω $V_{TT}$ | N7 | A12/BCN |
| A13 | G19 | SSTL15_II | 49.9Ω $V_{TT}$ | T3 | NC/A13 |
| A14 | F20 | SSTL15_II | 49.9Ω $V_{TT}$ | T7 | NC/A14 |
| BA0 | K17 | SSTL15_II | 49.9Ω $V_{TT}$ | M2 | BA0 |
| BA1 | L17 | SSTL15_II | 49.9Ω $V_{TT}$ | N8 | BA1 |
| BA2 | K18 | SSTL15_II | 49.9Ω $V_{TT}$ | M3 | BA2 |
| RAS# | K21 | SSTL15_II | 49.9Ω $V_{TT}$ | J3 | RAS# |
| CAS# | K22 | SSTL15_II | 49.9Ω $V_{TT}$ | K3 | CAS# |
| WE# | K19 | SSTL15_II | 49.9Ω $V_{TT}$ | L3 | WE# |
| CS# | - | - | 100Ω GND | L2 | CS# |
| RESET# | H18 | LVCMOS15 | 4.7kΩ GND | T2 | RESET# |
| CKE | F21 | SSTL15_II | 4.7kΩ GND | K9 | CKE |
| ODT | J22 | SSTL15_II | 49.9Ω $V_{TT}$ | K1 | ODT |
| DQ0 | R20 | SSTL15_II | ODT | E3 | DQ0 |
| DQ1 | R22 | SSTL15_II | ODT | F7 | DQ1 |
| DQ2 | P21 | SSTL15_II | ODT | F2 | DQ2 |

| Signal | DDR Bank A FPGA Pin | I/O Standard | Termination | Memory Device | |
|---|---|---|---|---|---|
| | | | | Pin | Name |
| DQ3 | P22 | SSTL15_II | ODT | F8 | DQ3 |
| DQ4 | L20 | SSTL15_II | ODT | H3 | DQ4 |
| DQ5 | L22 | SSTL15_II | ODT | H8 | DQ5 |
| DQ6 | M21 | SSTL15_II | ODT | G2 | DQ6 |
| DQ7 | M22 | SSTL15_II | ODT | H7 | DQ7 |
| DQ8 | T21 | SSTL15_II | ODT | D7 | DQ8 |
| DQ9 | T22 | SSTL15_II | ODT | C3 | DQ9 |
| DQ10 | U20 | SSTL15_II | ODT | C8 | DQ10 |
| DQ11 | U22 | SSTL15_II | ODT | C2 | DQ11 |
| DQ12 | W20 | SSTL15_II | ODT | A7 | DQ12 |
| DQ13 | W22 | SSTL15_II | ODT | A2 | DQ13 |
| DQ14 | Y21 | SSTL15_II | ODT | B8 | DQ14 |
| DQ15 | Y22 | SSTL15_II | ODT | A3 | DQ15 |
| LDQS | N20 | DIFF_SSTL15_II | ODT | F3 | LDQS |
| LDQS# | N22 | DIFF_SSTL15_II | ODT | G3 | LDQS# |
| UDQS | V21 | DIFF_SSTL15_II | ODT | C7 | UDQS |
| UDQS# | V22 | DIFF_SSTL15_II | ODT | B7 | UDQS# |
| LDM | N19 | SSTL15_II | ODT | E7 | LDM |
| UDM | P20 | SSTL15_II | ODT | D3 | UDM |
| CK | K20 | DIFF_SSTL15_II | 100Ω | J7 | CK |
| CK# | L19 | DIFF_SSTL15_II | | K7 | CK# |
| RZQ | F18 | SSTL15_II | 100Ω GND | - | - |
| ZIO | P19 | SSTL15_II | open | - | - |

Table 7-8 : DDR3 SDRAM Interface

For details regarding the DDR3 SDRAM interface, please refer to the DDR3 SDRAM Data Sheet and the Xilinx UG388: *Spartan-6 FPGA Memory Controller User Guide*.

### 7.8.2 SPI-Flash

The TXMC635 provides a Winbond W25Q32 32-Mbit serial Flash memory. This Flash is used as FPGA configuration source (default configuration source). The TXMC635 could be delivered with the W25Q32FV or the W25Q32JV.

**In contrast to the W25Q32FV EEPROM, the W25Q32JV EEPPROM does not support the QPI mode.**

After configuration, it is always accessible from the FPGA, so it also can be used for code or user data storage.

The SPI-EEPROM is connected via Quad (x4) SPI interface to the Spartan6 configuration interface.

| SPI-PROM Signal | Bank | $V_{CCO}$ | Pin | Description / Spartan6 |
|---|---|---|---|---|
| CLK | 2 | 3.3V | Y20 | Serial Clock (CCLK) |
| CS# | 2 | 3.3V | AA3 | Chip Select (CS0_B) |
| DI (bit0) | 2 | 3.3V | AB20 | Serial Data input (MOSI) / MISO[0] |
| DO (bit1) | 2 | 3.3V | AA20 | Serial Data output (DIN) / MISO[1] |
| WP# (bit2) | 2 | 3.3V | R13 | MISO[2] |
| HOLD# (bit3) | 2 | 3.3V | T14 | MISO[3] |

Table 7-9 : FPGA SPI-Flash Connections

### 7.8.3 I2C - EEPROM

The TXMC635 provides an Atmel AT24C04D (512x8) I2C-Compatible (2-wire) Serial EEPROM.

This EEPROM is used as ADC and DAC calibration data source. During factory test the analog I/O channel gain error and offset error are determined. For each device and each gain selection a 16 bit correction value is stored to the I2C EEPROM. These calibration data have been determined with TEWS test environment and build such a possible basis. If system specific calibration data are needed, the calibration of the entire system can be done by user and the I2C EEPROM could be used as a possible memory.

The I2C EEPROM is connected via 2-wire interface to User FPGA Spartan6. As usual for the I2C interface the two pins must be realized as open drain buffer.

| SPI-PROM Signal | Bank | $V_{CCO}$ | Pin | Description / Spartan6 |
|---|---|---|---|---|
| FPGA_SCL | 1 | 1.5V | R17 | Serial clock |
| FPGA_SDA | 1 | 1.5V | T17 | Serial data |

Table 7-10: FPGA I2C EEPROM Connections

For using the serial I2C interface between the USER FPGA (Spartan6) and the I2C EEPROM please see the Atmel AT24C04D data sheet which describes the serial communication process.

#### 7.8.3.1 I2C Calibration Data

There are two errors affecting the accuracy of the ADC and DAC that can be corrected using the factory calibrated calibration data. The correction values are obtained during factory calibration and are stored in an on-board I2C EEPROM as 2-complement 16 bit values in the range from -32768 to +32767. To achieve a higher accuracy, they are scaled to ¼LSB.

DAC Offset Error:

For the DAC, this is the data value that is required to produce a zero voltage output signal. This error is corrected by subtracting the offset from the DAC data value.

ADC Offset Error:

The offset error is the data value when converting with the input connected to its own ground in single-ended mode, or with shorted inputs in differential mode. This error is corrected by subtracting the known error from the reading.

DAC Gain Error:

The gain error is the difference between the ideal gain and the actual gain of the DAC. It is corrected by multiplying the DAC data value with a correction factor.

ADC Gain Error:

The gain error is the difference between the ideal gain and the actual gain of the programmable gain amplifier and the ADC. This error is corrected by multiplying the reading with a correction factor.

#### 7.8.3.2 ADC Calibration Data Values

The 32 ADC channels are realized with four ADAS3022 ADC devices. For each device and for each possible programmable gain selection (PGIA) respective values are stored.

| PGIA | | Absolute Input Voltage Range |
|---|---|---|
| 0b000 | 0x0 | ± 24.576V |
| 0b001 | 0x1 | ± 10.24V |
| 0b010 | 0x2 | ± 5.12V |
| 0b011 | 0x3 | ± 2.56V |
| 0b100 | 0x4 | ± 1.28V |
| 0b101 | 0x5 | ± 0.64V |
| 0b110 | 0x6 | not used |
| 0b111 | 0x7 | ± 20.48V |

Table 7-11: ADAS3022 Programmable gain selections

**For Programmable gain selections (PGIA) see also the ADAS3022 data sheet.**

| I2C EEPROM Address | Description | Size (Bit) |
|---|---|---|
| 0x000 | ADC Device 1 PGIA = 0x0 Offset Error High Byte | 8 |
| 0x001 | ADC Device 1 PGIA = 0x0 Offset Error Low Byte | 8 |
| 0x002 | ADC Device 1 PGIA = 0x0 Gain Error High Byte | 8 |
| 0x003 | ADC Device 1 PGIA = 0x0 Gain Error Low Byte | 8 |
| 0x004 | ADC Device 2 PGIA = 0x0 Offset Error High Byte | 8 |
| 0x005 | ADC Device 2 PGIA = 0x0 Offset Error Low Byte | 8 |
| 0x006 | ADC Device 2 PGIA = 0x0 Gain Error High Byte | 8 |
| 0x007 | ADC Device 2 PGIA = 0x0 Gain Error Low Byte | 8 |
| 0x008 | ADC Device 3 PGIA = 0x0 Offset Error High Byte | 8 |
| 0x009 | ADC Device 3 PGIA = 0x0 Offset Error Low Byte | 8 |
| 0x00A | ADC Device 3 PGIA = 0x0 Gain Error High Byte | 8 |
| 0x00B | ADC Device 3 PGIA = 0x0 Gain Error Low Byte | 8 |
| 0x00C | ADC Device 4 PGIA = 0x0 Offset Error High Byte | 8 |
| 0x00D | ADC Device 4 PGIA = 0x0 Offset Error Low Byte | 8 |
| 0x00E | ADC Device 4 PGIA = 0x0 Gain Error High Byte | 8 |
| 0x00F | ADC Device 4 PGIA = 0x0 Gain Error Low Byte | 8 |
| … | | |
| 0x07E | ADC Device 4 PGIA = 0x7 Gain Error High Byte | 8 |
| 0x07F | ADC Device 4 PGIA = 0x7 Gain Error Low Byte | 8 |

Table 7-12: ADC Calibration Data Values

### 7.8.3.3 DAC Calibration Data Values

There is one Offset Error value and one Gain Error value for each DAC channel. A differentiation of the correction value in relation to the set output range does not exist.

| I2C EEPROM Address | Description | Size (Bit) |
|---|---|---|
| 0x080 | DAC Channel 1 Offset Error High Byte | 8 |
| 0x081 | DAC Channel 1 Offset Error Low Byte | 8 |
| 0x082 | DAC Channel 1 Gain Error High Byte | 8 |
| 0x083 | DAC Channel 1 Gain Error Low Byte | 8 |
| 0x084 | DAC Channel 2 Offset Error High Byte | 8 |
| 0x085 | DAC Channel 2 Offset Error Low Byte | 8 |
| 0x086 | DAC Channel 2 Gain Error High Byte | 8 |
| 0x087 | DAC Channel 2 Gain Error Low Byte | 8 |
| 0x088 | DAC Channel 3 Offset Error High Byte | 8 |
| 0x089 | DAC Channel 3 Offset Error Low Byte | 8 |
| 0x08A | DAC Channel 3 Gain Error High Byte | 8 |
| 0x08B | DAC Channel 3 Gain Error Low Byte | 8 |
| 0x08C | DAC Channel 4 Offset Error High Byte | 8 |
| 0x08D | DAC Channel 4 Offset Error Low Byte | 8 |
| 0x08E | DAC Channel 4 Gain Error High Byte | 8 |
| 0x08F | DAC Channel 4 Gain Error Low Byte | 8 |
| 0x090 | DAC Channel 5 Offset Error High Byte | 8 |
| 0x091 | DAC Channel 5 Offset Error Low Byte | 8 |
| 0x092 | DAC Channel 5 Gain Error High Byte | 8 |
| 0x093 | DAC Channel 5 Gain Error Low Byte | 8 |
| 0x094 | DAC Channel 6 Offset Error High Byte | 8 |
| 0x095 | DAC Channel 6 Offset Error Low Byte | 8 |
| 0x096 | DAC Channel 6 Gain Error High Byte | 8 |
| 0x097 | DAC Channel 6 Gain Error Low Byte | 8 |
| 0x098 | DAC Channel 7 Offset Error High Byte | 8 |
| 0x099 | DAC Channel 7 Offset Error Low Byte | 8 |
| 0x09A | DAC Channel 7 Gain Error High Byte | 8 |
| 0x09B | DAC Channel 7 Gain Error Low Byte | 8 |
| 0x09C | DAC Channel 8 Offset Error High Byte | 8 |
| 0x09D | DAC Channel 8 Offset Error Low Byte | 8 |
| 0x09E | DAC Channel 8 Gain Error High Byte | 8 |
| 0x09F | DAC Channel 8 Gain Error Low Byte | 8 |

Table 7-13: DAC Calibration Data Values

For correcting the DAC output value the Coarse Gain Register, Fine Gain Register and Offset Register of the AD5764R could be used.

### 7.8.3.4 ADC Data Correction Formula

**Please use the total 16 bit data register value for the ADC correction formula.**

The basic formula for correcting any ADC reading for the TXMC635 (bipolar input voltage range) is:

$$Value = Reading \cdot \left(1 - \frac{Gain_{corr}}{131072}\right) - \frac{Offset_{corr}}{4}$$

*Value* is the corrected result.

*Reading* is the data read from the ADC Data Register.

$Gain_{corr}$ and $Offset_{corr}$ are the ADC correction factors from the Calibration Data ROM stored for each programmable gain factor.

The correction values are stored as two's complement 16 bit values in the range -32768 to 32767. For higher accuracy they are scaled to ¼ LSB.

**Floating point arithmetic or scaled integer arithmetic is necessary to avoid rounding error while computing above formula.**

### 7.8.3.5 DAC Data Correction Formula

The basic formula for correcting any DAC value is:

$$Data = Value \cdot \left(1 - \frac{Gain_{corr}}{131072}\right) - \frac{Offset_{corr}}{4}$$

*Value* is the desired DAC value.

*Data* is the corrected DAC value that must be sending to the DAC.

$Gain_{corr}$ and $Offset_{corr}$ are the DAC correction values from the Calibration Data ROM. They are stored separately for each of the 8 DAC channels.

The correction values are stored as two's complement byte wide values in the range from –32768 to +32767. For higher accuracy they are scaled to ¼ LSB.

**Floating point arithmetic or scaled integer arithmetic must be used to avoid rounding errors in computing above formula.**

# 7.9 Serial ADC Interface

The 32 analog inputs of the TXMC635 are realized with four ADAS3022 ADC devices. Each of these four SAR-ADCs has an 8 channel input MUX with integrated PGA. The eight analog input channels could be used as eight single-ended or four differential inputs.

The digital connection to the Sparten6 User-FPGA is realized via a Serial Peripheral Interface (SPI). In addition, each ADC device provides four additional control lines (RESET, Power-Down, BUSY and CNV) for conversion process control.

The selected ADC device provides on all analog input lines diodes for protection against electrostatic charging. Additional input line protection is supported by an ADG467 octal channel protection device.

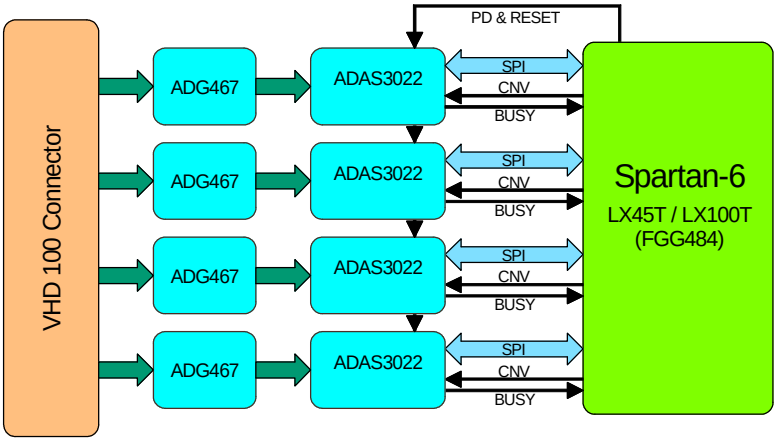



Figure 7-6 : Serial ADC Interface

Each ADC is connected via a dedicated SPI bus to the Spartan6 User-FPGA. Also the conversion control line CNV and the status line BUSY are connected individually to the Spartan6 User-FPGA. This allows reading all four ADCs simultaneously, thus exploiting the minimum conversion time. Only the control line Power-Down and RESET are shared by all four ADC devices.

| Signal | Bank | VCCO | Pin | Description |
|---|---|---|---|---|
| CS_ADC_[0]# | 0 | 3.3V | A19 | ADC active low chip select |
| DIN_ ADC[0] | 0 | 3.3V | A18 | ADC serial data input |
| SCK_ ADC[0] | 0 | 3.3V | B18 | ADC serial clock input |
| SDO_ ADC[0] | 0 | 3.3V | C19 | ADC serial data output |
| BUSY_ ADC[0] | 1 | 1.5V | R15 | ADC busy output |
| CNV_ ADC[0] | 0 | 3.3V | D19 | ADC conversion input |
| CS_ADC_[1]# | 0 | 3.3V | G9 | ADC active low chip select |
| DIN_ ADC[1] | 0 | 3.3V | H12 | ADC serial data input |
| SCK_ ADC[1] | 0 | 3.3V | G11 | ADC serial clock input |
| SDO_ ADC[1] | 0 | 3.3V | F10 | ADC serial data output |
| BUSY_ ADC[1] | 1 | 1.5V | R16 | ADC busy output |
| CNV_ ADC[1] | 0 | 3.3V | C18 | ADC conversion input |
| CS_ADC_[2]# | 3 | 3.3V | K6 | ADC active low chip select |
| DIN_ ADC[2] | 3 | 3.3V | M8 | ADC serial data input |
| SCK_ ADC[2] | 3 | 3.3V | M7 | ADC serial clock input |
| SDO_ ADC[2] | 3 | 3.3V | L6 | ADC serial data output |
| BUSY_ ADC[2] | 1 | 1.5V | V19 | ADC busy output |
| CNV_ ADC[2] | 3 | 3.3V | P7 | ADC conversion input |
| CS_ADC_[3]# | 3 | 3.3V | L1 | ADC active low chip select |
| DIN_ ADC[3] | 3 | 3.3V | N4 | ADC serial data input |
| SCK_ ADC[3] | 3 | 3.3V | H8 | ADC serial clock input |
| SDO_ ADC[3] | 3 | 3.3V | P3 | ADC serial data output |
| BUSY_ ADC[3] | 1 | 1.5V | P17 | ADC busy output |
| CNV_ ADC[3] | 3 | 3.3V | K8 | ADC conversion input |
| RESET_ADC | 1 | 1.5V | A20 | Shared Asynchronous Reset |
| PD_ADC | 1 | 1.5V | B20 | Shared Power-Down |

Table 7-14: ADC Interface Connections

For using the serial SPI interface between the USER FPGA (Spartan6) and one of the four ADAS3022 ADC devices please use the ADAS3022 data sheet which describes the communication process.

# 7.10 Serial DAC Interface

The 8 analog outputs of the TXMC635 are realized with two 16-bit quad digital-to-analog converter devices (AD5764R). Each DAC device has four output channels with a nominal full-scale range of ±10V. The AD5764R provides integrated output amplifiers, reference buffers and proprietary power-up/power-down control circuitry.

The Operational Amplifier ADTL082 is used to optimize the driver properties of analog Output interface.

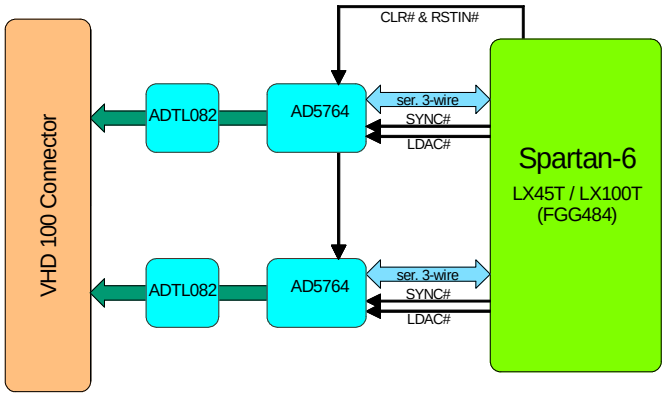



Figure 7-7 : Serial DAC Interface

The digital connection to the Sparten6 User-FPGA is realized via a 3-wire serial interface that operates at clock rates of up to 30MHz. This interface is compatible with SPI, QSPU and MICROWIRE standards. In addition, each device provides four more typical DAC control lines (SYNC#, LDAC#, CLR# and RSTIN#) to control the conversion process.

Both DAC devices are connected via a dedicated serial interface to the Spartan6 User-FPGA. Also the conversion control signals SYNC# and LDAC# are connected individually to the Spartan6 User-FPGA.

| Signal | Bank | VCCO | Pin | Description |
|---|---|---|---|---|
| SYNC#_DAC[0] | 0 | 3.3V | F15 | |
| SCLK_DAC[0] | 0 | 3.3V | H10 | |
| SDIN_DAC[0] | 0 | 3.3V | F16 | |
| SDO_DAC[0] | 0 | 3.3V | E16 | |
| LDAC#_DAC[0] | 0 | 3.3V | F14 | |
| SYNC#_DAC[1] | 0 | 3.3V | A17 | |
| SCLK_DAC[1] | 0 | 3.3V | H14 | |
| SDIN_DAC[1] | 0 | 3.3V | G15 | |
| SDO_DAC[1] | 0 | 3.3V | C17 | |
| LDAC#_DAC[1] | 0 | 3.3V | G16 | |
| CLR#_DAC | 0 | 3.3V | H13 | |
| RSTIN#_DAC | 0 | 3.3V | H11 | |

Table 7-15: Serial DAC Interfacer

For using the serial SPI interface between the USER FPGA (Spartan6) and one of the two AD5764 DAC devices please refer to the AD5764 data sheet which describes the communication process.

# 7.11 Serial Number Allocation

The TXMC635 Module Serial Number is stored on-board the module, and can be read on both FPGA devices. The Configuration FPGA (MachXO2) provides a Serial Number Register in the local register space.

For the User FPGA (Spartan6) an I2C Master interface is required to read the serial number from the Configuration FPGA. For this purpose the Configuration FPGA provides an I2C slave interface.

| Signal | Bank | $V_{CCO}$ | Pin | Description |
|---|---|---|---|---|
| FPGA_SCL | 1 | 1.5V | R17 | Serial Clock Output<br>A negative edge clock data out. |
| FPGA_SDA | 1 | 1.5V | P18 | Bisectional Serial Data |

Table 7-16: User FPGA I2C Interface to Configuration FPGA

The Configuration I2C Interface provides only one readable register. The Serial Number Register is a 32 bit wide read only register. The Slave Address of the Serial Number Register is 0b1010101.

The support frequencies are between 100kHz up to 400kHz.

| Bit | Symbol | Description | Access | Reset Value |
|---|---|---|---|---|
| 31:0 | S_NUMBER | The value is the unique serial number of each TXMC635 module | R | - |

Table 7-17: TXMC635 Serial Number

Example: 0x008F_DD0F => SNo.: 9428239

## 7.11.1 Device Addressing and Operation

The TXMC635 Configuration FPGA uses a standard 7 bit Slave Address. The eighth bit of the slave address is the Read/write operation select bit.

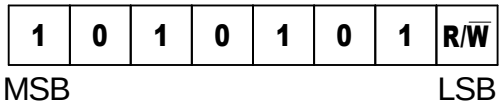



Figure 7-8 : Configuration FPGA Slave Address

TXMC635 Configuration FPFA I2C Slave typically Start and Stop condition

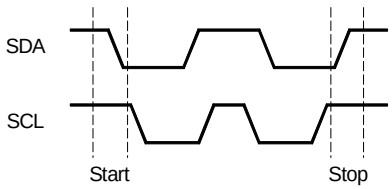



Figure 7-9 : Configuration FPGA Start and Stop condition

TXMC635 Configuration FPFA I2C Slave Output Acknowledge

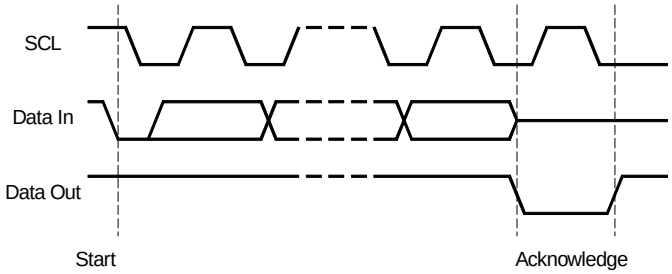



Figure 7-10 : Configuration FPGA Output Acknowledge

## 7.11.2 Read Operation

The TXMC635 Configuration FPGA provides only one 32 bit register which could be read from User FPGA via a I2C Interface.

The read operation starts with a I2C start condition followed by a 7 bit slave address. The read/write bit in the device address byte is set to one. The configuration FPGA acknowledged the address and began to transmit all four data byte of the TXM633 Serial Number Register. Each byte must be acknowledged. The sequence must be completed with a stop condition by the User FPGA.

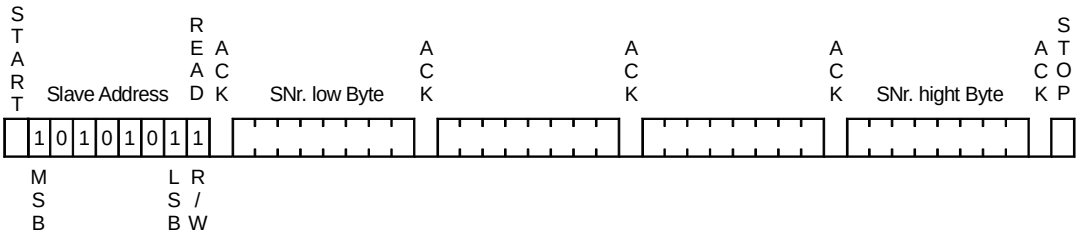



Figure 7-11 : Configuration FPGA Slave Access

## 7.11.3 Write Operation

A I2C write operation is not implemented.

# 7.12 I/O Pull Configuration

Each TTL I/O Line has a 4k7 Pull-Resistor. The 48 I/O Lines are arranged in three groups which can be configured for 3.3V pull-up, 5V pull-up or pull-down. In addition, the Pull-Resistors can float.

If the Pull-Resistors float, the user should keep in mind that the 16 I/O Lines of the group are connected via their Pull-Resistors.

The normal behaviour is that the User FPGA code controls the I/O Pull Configuration depending on the User FPGA I/O Function.

The User FPGA (Spartan6) IO_PULL Interface configuration signals are connected to three analog multiplexers via the Configuration FPGA. With these multiplexers the desired pull resistor reference can be adjusted directly from the User FPGA. In total there are six I/O lines controlling the three analog multiplexer regarding the I/O Pull Voltage. The user must ensure that valid signals are generated on all six control lines. The control lines must be driven low for a low level and must be set to High-Z for a high level.

| CNT **Lines** | **I/O Lines** | **Description** | **Spartan6 Pins** |
|---|---|---|---|
| IO_PULL[5:4] | IO_31 : IO_47 | 0bZZ : pull-down<br>0bZ0 : pull-up to 3.3V<br>0b0Z : pull-up to 5V<br>0b00 : No pull-up or pull-down | M18, M17 |
| IO_PULL[3:2] | IO_16 : IO_30 | | P16, N16 |
| IO_PULL[1:0] | IO_00 : IO_15 | | T20, U19 |

Table 7-18: I/O Pull Configuration

The Configuration FPGA offers an additional option for setting the I/O Pull. Using the User FPGA Configuration Control/Status Register the control can be taken to the Configuration FPGA. Use the I/O Pull Resistor Configuration Register of the Configuration FPGA to set the desired pull resistor reference.

# 7.13 User GPIO

The TXMC635 has some general purpose I/O and debug signals connected to User FPGA Bank 1. The required signaling standard is LVCMOS15, due to the Memory Controller Block usage.

Two pins of the FPGA are routed to the Debug Connector for use as a debug interface (UART). Note, that this is not a real RS232 interface. A RS232 transceiver or USB-UART that can work with 1.5V I/O voltage should connect to these signals such as TEWS TA900.

A general purpose I/O Signal is also connected to the Debug Connector. When used with the TEWS TA900, this signal is connected to a Push button and must be configured as FPGA input.

Also two free user programmable LEDs are connected to the User FPGA Bank 1.

| Signal | Bank | $V_{CCO}$ | Pin | Description |
|---|---|---|---|---|
| GPIO_LED0 | 1 | 1.5V | M16 | 4x green on-board LEDs |
| GPIO_LED1 | | | N15 | |
| FPGA_BUT | 1 | 1.5V | T18 | General Purpose User I/O |
| FPGA_RXD | 1 | 1.5V | T17 | Serial Debug Interface is accessible via TEWS debug-connector. |
| FPGA_TXD | 1 | 1.5V | T19 | |

Table 7-19: FPGA General Purpose I/O

# 7.14 On-Board Indicators

The TXMC635 provides a couple of board-status LEDs as shown below. These include Power-Good and FPGA configuration status indications as well as four general purpose LEDs.

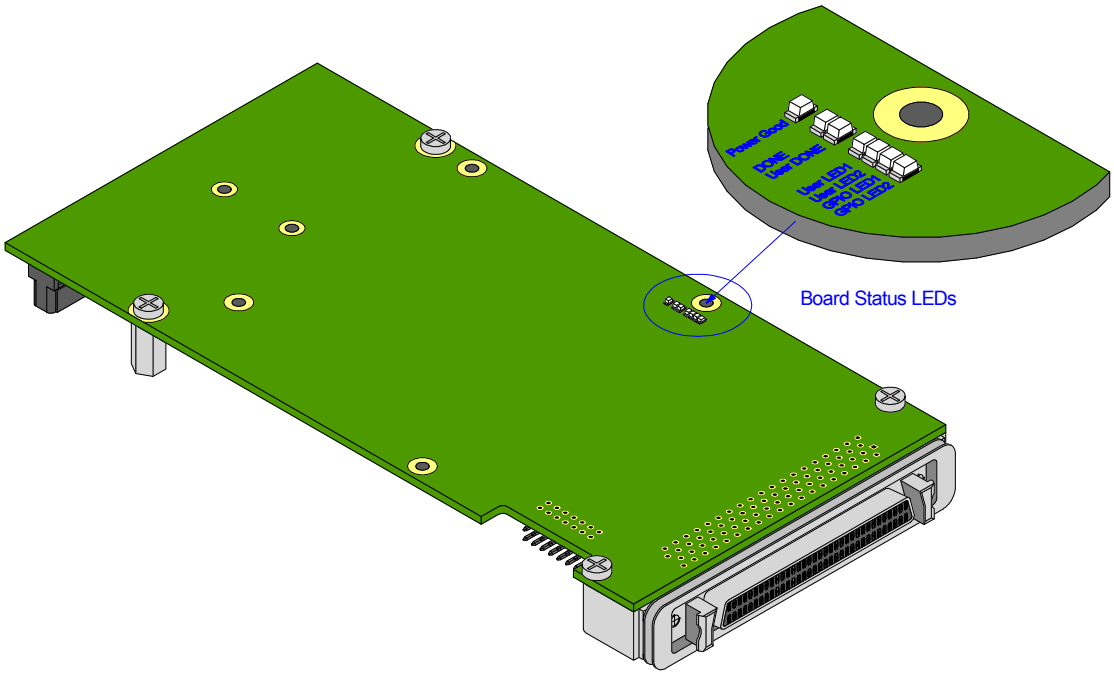



Figure 7-12 : On Board Indicators

| LED | Color | State | | Description |
|---|---|---|---|---|
| Power Good | Green | off | On-Board Power Supplies are not ok | Power Good<br>Signal for all on-board power supplies. |
| | | on | On-Board Power Supplies are all ok | |
| DONE | Green | off | Device is not configured | Configuration FPGA DONE-Pin LED<br>(MachXO2) Indicates successful FPGA configuration |
| | | on | Device is completely configured | |
| User DONE | Green | off | Device is not configured | User FPGA DONE-Pin LED (Spartan6)<br>Indicates successful FPGA configuration |
| | | | Device is completely configured | |
| USER LED1 | Green | - | | Design dependent, can be controlled by the User FPGA.<br>Refer to chapter “User-GPIO” |
| USER LED2 | Green | | | |
| GPIO LED1 | Green | off | General State is not IDLE | Configuration FPGA depends. |
| | | on | General State is IDLE | |
| GPIO LED2 | Green | off | PCI Reset is active | |
| | | on | PCI Reset is inactive | |

Table 7-20: Board-Status and User LEDs

## 7.15 Thermal Management

Power dissipation is design dependent. Main factors are device utilization, frequency and GTP-transceiver usage. Use the Xilinx XPower Estimator (XPE) or XPower Analyzer to determine whether additional cooling requirements such as forced air cooling apply. Forced air cooling is recommended during operation.

The TXMC635 has a heat sink mounted on the Spartan-6 FPGA.

# 8 Design Help

## 8.1 Analog Inputs

The 32 Analog inputs of the TXMC635 are connected to a total of four multiplexed Analog Data Acquisition Systems (ADAS3022). Each ADAS3022 is connected to 8 analog inputs. The signal path from the Input Connector pins to the ADC inside the ADAS3022 is shown in the following figure.

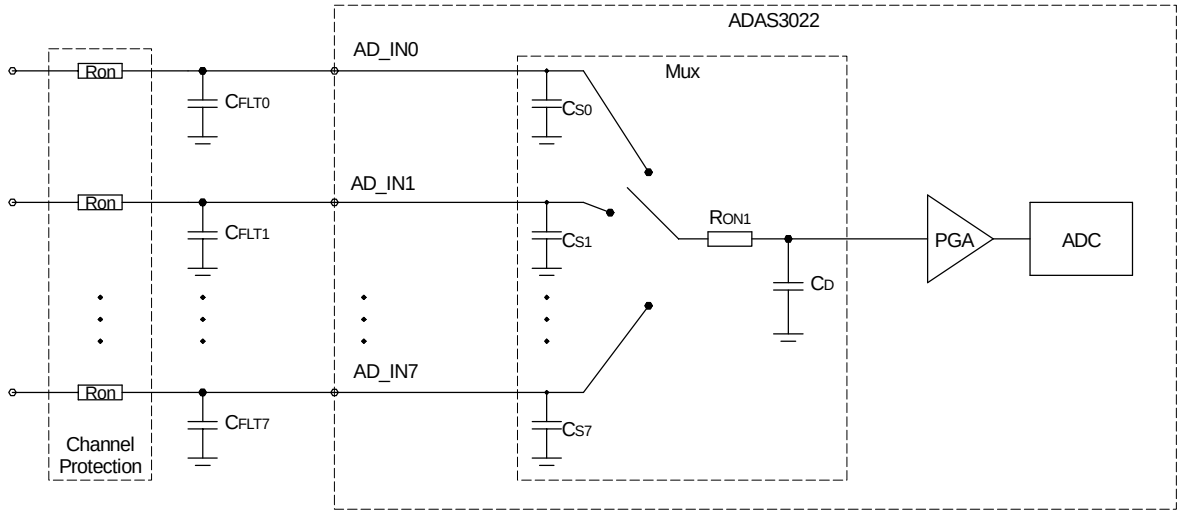



Figure 8-1 : Analog input path

The on-resistance of the Channel Protection device Ron along with the filter capacitor $C_{FLT}$ forms a passive 1st order low-pass filter on each analog input. The primary purpose of this input filter is to reject noise, generated by e.g. switching digital I/O lines, that couples on the analog input wires.

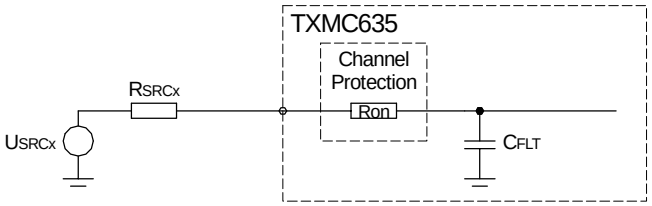



Figure 8-2 : Analog Input filter

The 3dB cutoff frequency of this filter is app. 480kHz for a low impedance source. As the source impedance $R_{SRCx}$ is in series with the filter resistor $R_{on}$, it influences the resulting filter cutoff frequency.

Inside the ADAS3022, there is an analog multiplexer (Mux) in front of the programmable gain amplifier (PGA). The output capacitance $C_D$ of the Mux is recharged via the on-resistance $R_{ON1}$ each time the Mux switches. This recharging generates a so called “kickback-pulse” on the analog input. For correct measuring, this kickback-pulse must settle before the ADC samples the new input channel.

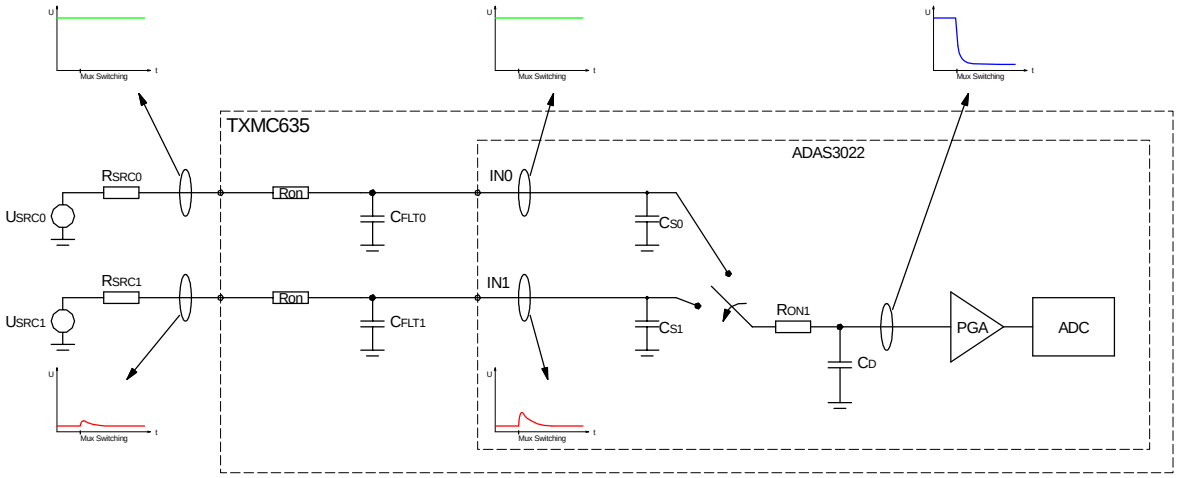



Figure 8-3 : ADAS3022 Mux Switching

Due to the Mux’s “break before make” structure, the kickback-pulse is only visible on the analog input the mux switches to.

# 8.2 Example Design

User applications for the TXMC635 may be developed by using the TXMC635 FPGA Example Application design.

TEWS offers this FPGA Example design as a well-documented example basic example. It includes an .ucf file with all necessary pin assignments and basic timing constraints. The example design covers the main functionalities of the TXMC635. It implements a PCIe endpoint with interrupt support, register mapping, DDR3 memory access and basic I/O functions. It comes as a Xilinx ISE 14.7 project with source code and as a ready-to-download bit stream. This example design can be used as a starting point for own projects.

The TXMC635 FPGA Example Application design can be developed using the design software ISE Project Navigator (ISE) and Embedded Development Kit (EDK). IDE versions are 14.7. Licenses for both design tools are required.

For TXMC635 FPGA Example Application design see also the included User Manual.

# 9 Installation

## 9.1 I/O Interface

### 9.1.1 TTL I/O Interface

Each of the 48 TTL I/O lines is realized with a 74LVC2G241 dual buffer as an interface to the FPGA pins. The logic levels of the buffers are TTL compatible, meaning that the minimum high level is 2.0V and the maximum low level is 0.8V. The nominal output high voltage is 3.3V.

The buffer outputs are followed by 47Ω serial resistors for signal integrity reasons. The 4.7kΩ pull-resistors guaranty a TTL compatible logic level when outputs are tristate and not driven externally.

As an option the pull-up voltage can be set to 5V by an analog multiplexer to (weakly) drive a higher voltage than 3.3V by setting the output to tristate. This means, instead of toggling the corresponding bit of the output register, the output enable register bit is set to 0 for an output high level or 1 to pull the output low (the OUT_REG bit is '0'). For example when connecting to a standard 5V CMOS logic input (not TTL compatible levels), a high level of minimum 3.5V is required.

A second option is "set the pull-voltage to GND" to build pull-down functionality. This means, instead of toggling the corresponding bit of the output register, the output enable register bit is set to 0 for an output low level or 1 to drive the output high (the OUT_REG bit must be '1').

Please note, that the pull-up or pull-down resistor can only drive high impedance inputs.

A TVS array protects against ESD shocks.

See the following figure for more information of the TTL I/O circuitry.

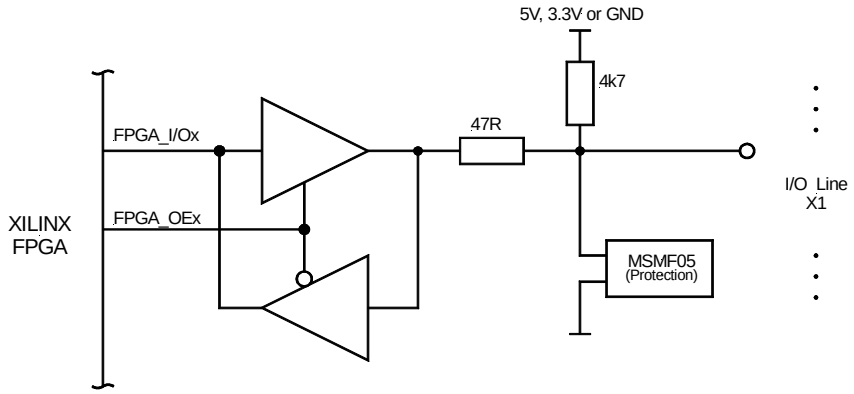



Figure 9-1 : TTL I/O Interface

**Please note, that the length (and consequently the capacitance) of a flat cable, connected to the TXMC635 module, should be kept as short as possible to prevent large cross talk.**

**To reduce the cross talk on the TXMC635, not all 48 I/O lines should be switched at the same time. For example, the output lines should be switched in groups of 8 signals in steps of 12ns, meaning that after about 100ns the switching process is completed.**

Each I/O Line has a 4k7 Pull-Resistor. These Pull-Resistors can be configured as 3.3V pull-up, 5V pull-up or pull-down. In addition, the Pull-Resistors can float.

If the Pull-Resistors float, the user should keep in mind that the I/O Lines of one group are connected via their Pull-Resistors.

Pull-Resistor configuration must be set with User FPGA code.

### 9.1.2 Back I/O Interface

P14 Back I/O Pins of the TXMC635 are direct routed to the Spartan6 FPGA. The I/O functions of these FPGA pins are directly dependent on the configuration of the FPGA.

The Spartan6 VCCO voltage is set to 3.3V, so only the 3.3V I/O standards LVCMOS33, LVTTL33 and LVDS_33 are possible for using on TXMC635 back I/O interface.

### 9.1.3 ADC

All analog inputs lines are routed through an over voltage protection devices which protect the ADAS3022 ADC inputs against fault voltage above ±30V.

| | |
|---|---|
| **Protection** | ±40 V overvoltage protection |
| **Input Impedance** | 500 MΩ |
| **Common-Mode Input Range** | ±24.576 V |

Table 9-1 : ADC Electrical Interface

The ADAS3022 can be configured to use the analog input pins as Single Ended Inputs. In addition, every "±" input pin pair can be configured as one Differential Input.

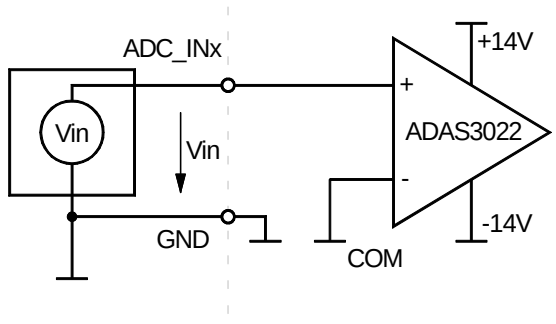
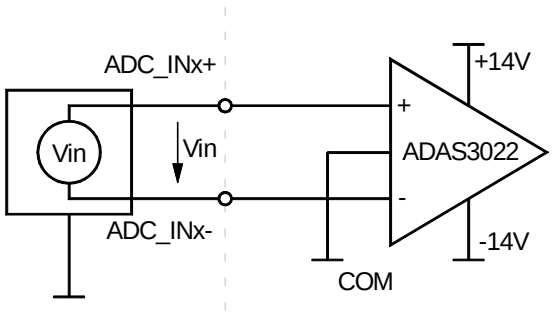
| **Single-Ended ADC Input** | **Differential ADC Input** |
|---|---|
|  |  |
| Maximum Vin = ±13.5V | Maximum Vin = ±24.576V |

Table 9-2 : ADC Input Schemes

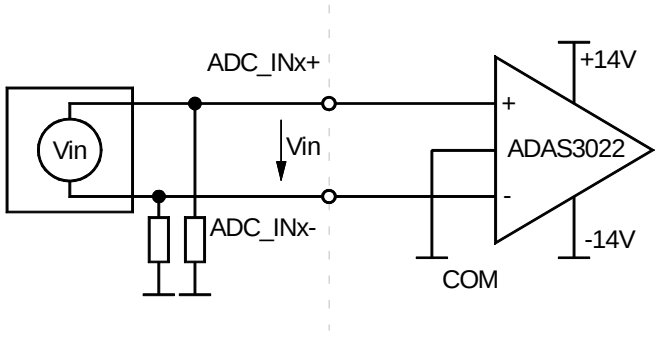
Due to the maximum input voltage (Vin) at each ADC input, not every programmable gain selection (PGIA) makes sense. For a Single-Ended input a maximum ADC setting of PGIA = ±10.24V is useful. If the inputs are used as differential all ADC PGIA settings are possible.

**If signals without a ground reference should be connected, connect ADC Lines to GND with a resistor to prevent the signal source to float out of the ADC's common-mode range.**

**Unused ADC channels should be connected to AGND.**

If signals without a ground reference should be connected, connect VIN+ and VIN- to GND with a resistor to prevent the signal source from floating out of the ADC's common-mode range. In most cases, a weak resistor to ground at the VIN- connection suffices.

**Differential without Ground Reference**




Maximum Vin = ±24.576V

Table 9-3 : ADC Input without Ground Reference

## 9.1.4 DAC

All analog outputs are routed through an operational amplifier in order to drive a higher capacitive load.

| | |
|---|---|
| **DC Output Impedance** | 0.25 Ω |
| **Load** | 1 kΩ |
| **Capacitive Load** | 1000 pF |

Table 9-4 : DAC Electrical Interface

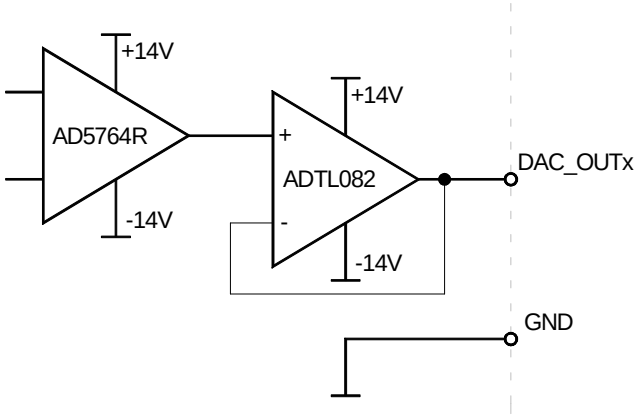



Figure 9-2 : DAC Output Interface

## 9.2 FPGA Debug Connector

The Debug Connector (X3) of the TXMC635 can be used to connect a debug adapter, if necessary. The debug adapter must be connected to the TXMC635 prior to XMC-Carrier installation. It is recommended to use the TEWS TA900 Debug Adapter.

The Debug Connector provides three logical interfaces: JTAG, FPGA-UART and one General Purpose User Signal (USER_BUT).

- The JTAG interface consists of the signals TDI, TDO, TMS, TCK, uses 3.3V I/O voltage, and can run with up to 6 MHz.
- The FPGA-UART consists of Rx and Tx and uses 1.5V I/O voltage. Communication settings depend on the FPGA programming.
- The General Purpose User Signal uses 1.5V I/O voltage. When used with the TEWS TA900, this signal is connected to a Push button on the TEWS TA900 and must be configured as FPGA input.

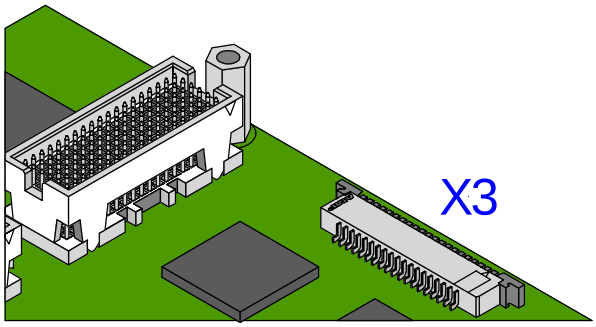



Figure 9-3 : Debug Connector X3

### 9.2.1 Connecting TA900 to TXMC635 Debug Connector

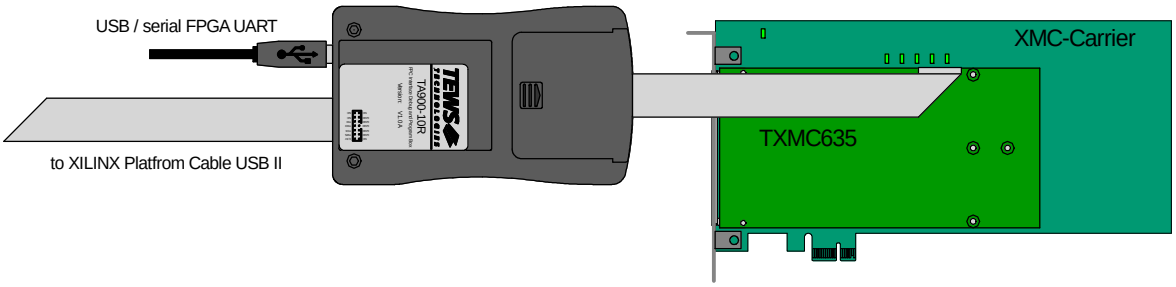



Figure 9-4 : 9.2.1 Connecting TA900 to TXMC635 Debug Connector

## 9.3 FPGA JTAG Connector

The FPGA JTAG connector X2 lets the user directly connect a JTAG interface cable to the on-board User FPGA JTAG chain, e.g. for FPGA read back and real-time debugging of the User FPGA design (using Xilinx “ChipScope”).

A through hole, right angle 90° connector with 7 x 2 pins and 2 mm pitch is mounted (Molex 0877601416 or compatible).

**With a mounted 2 mm pitch flat cable this is of cause a violation of the maximum component height given by the CMC specification, be sure that there is enough space to carrier board.**

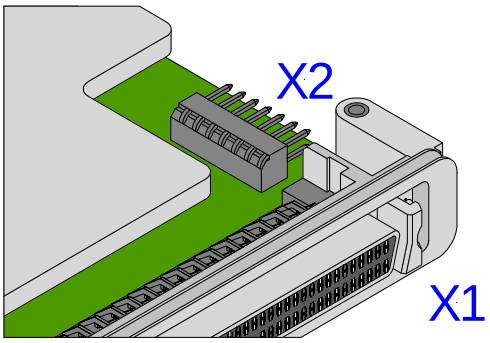



Figure 9-5 : FPGA JTAG Connector X2

# 10 Pin Assignment – I/O Connector

## 10.1 Overview

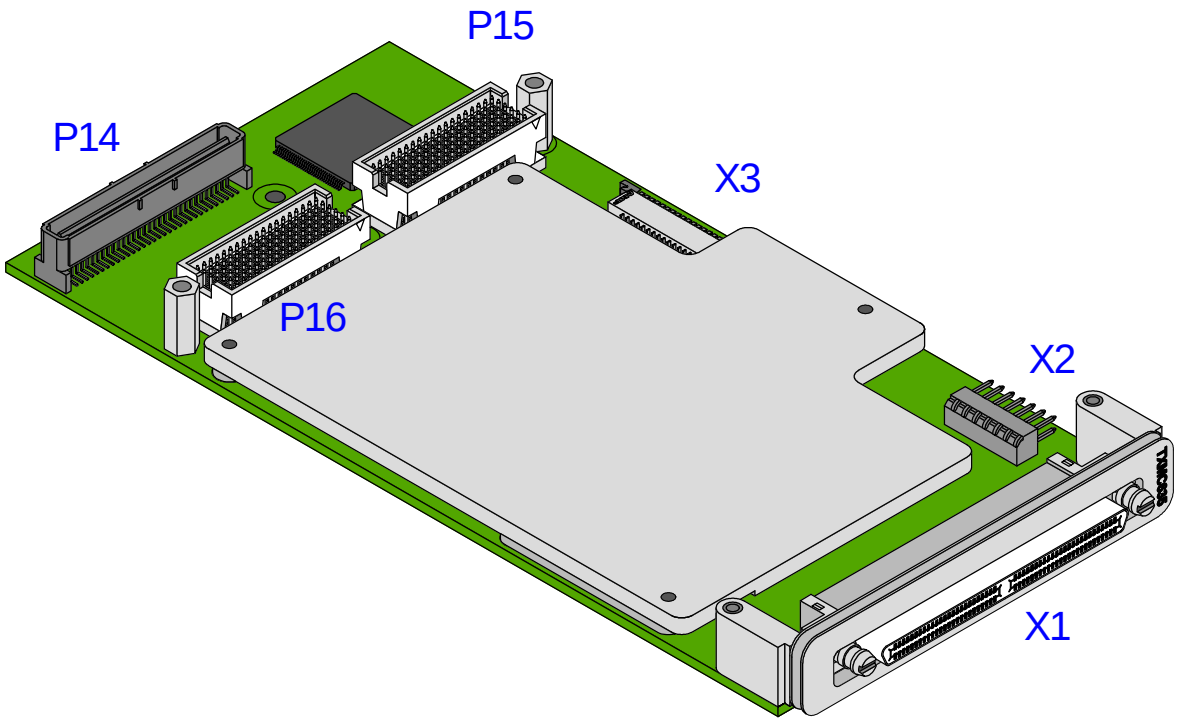



Figure 10-1 : Pin Assignment Overview

## 10.2 X1 Front Panel I/O Connector

### 10.2.1 Connector Type

| | |
|---|---|
| **Pin-Count** | 100 |
| **Connector Type** | VHD100 - 0.8mm Pitch Connector |
| **Source & Order Info** | Honda HDRA-EC100LFDT-SL+ |

### 10.2.2 Pin Assignment

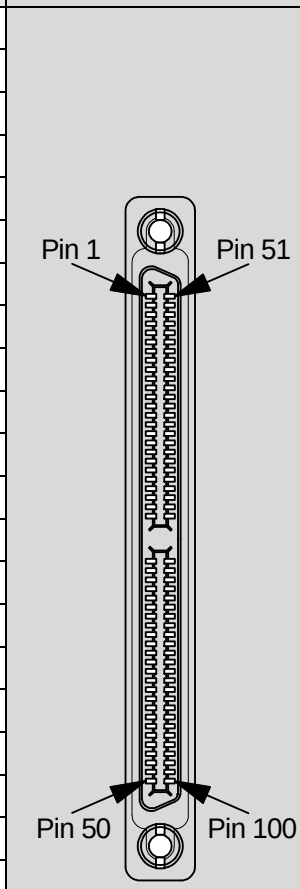
| Pin | I/O | Connector View | Pin | I/O |
|---|---|---|---|---|
| 1 | IO0 |  | 51 | IO24 |
| 2 | IO1 | | 52 | IO25 |
| 3 | IO2 | | 53 | IO26 |
| 4 | IO3 | | 54 | IO27 |
| 5 | IO4 | | 55 | IO28 |
| 6 | IO5 | | 56 | IO29 |
| 7 | IO6 | | 57 | IO30 |
| 8 | IO7 | | 58 | IO31 |
| 9 | IO8 | | 59 | IO32 |
| 10 | IO9 | | 60 | IO33 |
| 11 | IO10 | | 61 | IO34 |
| 12 | IO11 | | 62 | IO35 |
| 13 | IO12 | | 63 | IO36 |
| 14 | IO13 | | 64 | IO37 |
| 15 | IO14 | | 65 | IO38 |
| 16 | IO15 | | 66 | IO39 |
| 17 | IO16 | | 67 | IO40 |
| 18 | IO17 | | 68 | IO41 |
| 19 | IO18 | | 69 | IO42 |
| 20 | IO19 | | 70 | IO43 |
| 21 | IO20 | | 71 | IO44 |
| 22 | IO21 | | 72 | IO45 |
| 23 | IO22 | | 73 | IO46 |
| 24 | IO23 | | 74 | IO47 |
| 25 | GND | | 75 | GND |



Table 10-1: Pin Assignment Front Panel I/O Connector X1 digital Part

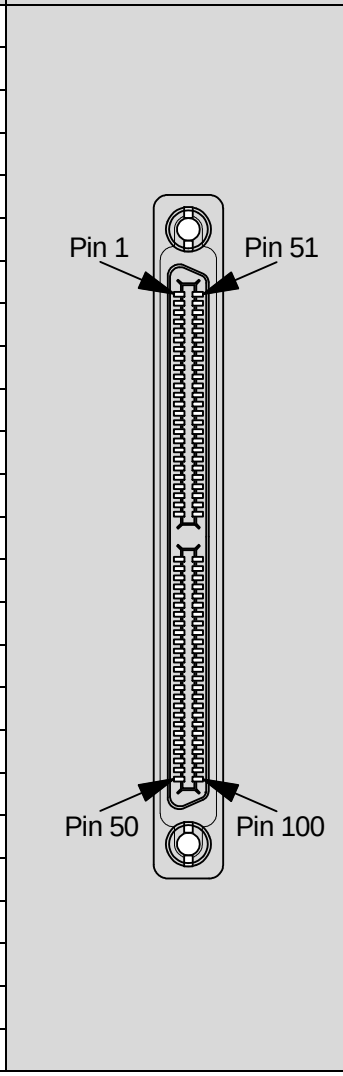
| Pin | I/O single-ended | I/O differential | Connector View | Pin | I/O single-ended | I/O differential |
|---|---|---|---|---|---|---|
| 26 | GND | |  | 76 | GND | |
| 27 | DAC_OUT1 | | | 77 | DAC_OUT5 | |
| 28 | GND | | | 78 | GND | |
| 29 | DAC_OUT2 | | | 79 | DAC_OUT6 | |
| 30 | GND | | | 80 | GND | |
| 31 | DAC_OUT3 | | | 81 | DAC_OUT7 | |
| 32 | GND | | | 82 | GND | |
| 33 | DAC_OUT4 | | | 83 | DAC_OUT8 | |
| 34 | GND | | | 84 | GND | |
| 35 | ADC_IN1 | ADC_IN1+ | | 85 | ADC_IN17 | ADC_IN9+ |
| 36 | ADC_IN2 | ADC_IN1- | | 86 | ADC_IN18 | ADC_IN9- |
| 37 | ADC_IN3 | ADC_IN2+ | | 87 | ADC_IN19 | ADC_IN10+ |
| 38 | ADC_IN4 | ADC_IN2- | | 88 | ADC_IN20 | ADC_IN10- |
| 39 | ADC_IN5 | ADC_IN3+ | | 89 | ADC_IN21 | ADC_IN11+ |
| 40 | ADC_IN6 | ADC_IN3- | | 90 | ADC_IN22 | ADC_IN11- |
| 41 | ADC_IN7 | ADC_IN4+ | | 91 | ADC_IN23 | ADC_IN12+ |
| 42 | ADC_IN8 | ADC_IN4- | | 92 | ADC_IN24 | ADC_IN12- |
| 43 | ADC_IN9 | ADC_IN5+ | | 93 | ADC_IN25 | ADC_IN13+ |
| 44 | ADC_IN10 | ADC_IN5- | | 94 | ADC_IN26 | ADC_IN13- |
| 45 | ADC_IN11 | ADC_IN6+ | | 95 | ADC_IN27 | ADC_IN14+ |
| 46 | ADC_IN12 | ADC_IN6- | | 96 | ADC_IN28 | ADC_IN14- |
| 47 | ADC_IN13 | ADC_IN7+ | | 97 | ADC_IN29 | ADC_IN15+ |
| 48 | ADC_IN14 | ADC_IN7- | | 98 | ADC_IN30 | ADC_IN15- |
| 49 | ADC_IN15 | ADC_IN8+ | | 99 | ADC_IN31 | ADC_IN16+ |
| 50 | ADC_IN16 | ADC_IN8- | | 100 | ADC_IN32 | ADC_IN16- |



Table 10-2: Pin Assignment Front Panel I/O Connector X1 analog Part

# 10.3 Back I/O XMC Connector P14

## 10.3.1 Connector Type

| | |
|---|---|
| **Pin-Count** | 64 |
| **Connector Type** | 64 pol. Mezzanine SMD Connector |
| **Source & Order Info** | Molex – 71436-2864 or compatible |

## 10.3.2 Pin Assignment

| Pin | differential I/O | | Pin | differential I/O |
|---|---|---|---|---|
| 1 | BACK_IO0+ | | 33 | BACK_IO16+ |
| 2 | BACK_IO0- | | 34 | BACK_IO16- |
| 3 | BACK_IO1+ | | 35 | BACK_IO17+ |
| 4 | BACK_IO1- | | 36 | BACK_IO17- |
| 5 | BACK_IO2+ | | 37 | BACK_IO18+ |
| 6 | BACK_IO2- | | 38 | BACK_IO18- |
| 7 | BACK_IO3+ | | 39 | BACK_IO19+ |
| 8 | BACK_IO3- | | 40 | BACK_IO19- |
| 9 | BACK_IO4+ | | 41 | BACK_IO20+ |
| 10 | BACK_IO4- | | 42 | BACK_IO20- |
| 11 | BACK_IO5+ | | 43 | BACK_IO21+ |
| 12 | BACK_IO5- | | 44 | BACK_IO21- |
| 13 | BACK_IO6+ | | 45 | BACK_IO22+ |
| 14 | BACK_IO6- | | 46 | BACK_IO22- |
| 15 | BACK_IO7+ | | 47 | BACK_IO23+ |
| 16 | BACK_IO7- | | 48 | BACK_IO23- |
| 17 | BACK_IO8+ | | 49 | BACK_IO24+ |
| 18 | BACK_IO8- | | 50 | BACK_IO24- |
| 19 | BACK_IO9+ | | 51 | BACK_IO25+ |
| 20 | BACK_IO9- | | 52 | BACK_IO25- |
| 21 | BACK_IO10+ | | 53 | BACK_IO26+ |
| 22 | BACK_IO10- | | 54 | BACK_IO26- |
| 23 | BACK_IO11+ | | 55 | BACK_IO27+ |
| 24 | BACK_IO11- | | 56 | BACK_IO27- |
| 25 | BACK_IO12+ | | 57 | BACK_IO28+ |
| 26 | BACK_IO12- | | 58 | BACK_IO28- |
| 27 | BACK_IO13+ | | 59 | BACK_IO29+ |
| 28 | BACK_IO13- | | 60 | BACK_IO29- |

| Pin | differential I/O | | Pin | differential I/O |
|---|---|---|---|---|
| 29 | BACK_IO14+ | | 61 | BACK_IO30+ |
| 30 | BACK_IO14- | | 62 | BACK_IO30- |
| 31 | BACK_IO15+ | | 63 | BACK_IO31+ |
| 32 | BACK_IO15- | | 64 | BACK_IO31- |

Figure 10-2: Pin Assignment P14 Back I/O Connector TXMC635

# 10.4 P16 Back I/O Connector

## 10.4.1 Connector Type

| Pin-Count | 114 |
|---|---|
| Connector Type | XMC Connector 114-pol Male |
| Source & Order Info | K39400885<br>Samtec - ASP-105885-01 |

## 10.4.2 Pin Assignment

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| **1** | Tx 0+ | Tx 0- | - | Tx 1+ | Tx 1- | - |
| **2** | GND | GND | - | GND | GND | - |
| **3** | Tx 2+ | Tx 2- | - | - | - | - |
| **4** | GND | GND | - | GND | GND | - |
| **5** | - | - | - | - | - | - |
| **6** | GND | GND | - | GND | GND | - |
| **7** | - | - | - | - | - | - |
| **8** | GND | GND | - | GND | GND | - |
| **9** | Reserved | Reserved | - | Reserved | Reserved | - |
| **10** | GND | GND | - | GND | GND | - |
| **11** | Rx 0+ | Rx 0- | - | Rx 1+ | Rx 1- | - |
| **12** | GND | GND | - | GND | GND | - |
| **13** | Rx 2+ | Rx 2- | - | - | - | - |
| **14** | GND | GND | - | GND | GND | - |
| **15** | - | - | - | - | - | - |
| **16** | GND | GND | - | GND | GND | - |
| **17** | - | - | - | - | - | - |
| **18** | GND | GND | - | GND | GND | - |
| **19** | - | - | - | - | - | - |

Figure 10-3: Pin Assignment P16 Back I/O Connector TXMC635-xxR

# 10.5 X2 JTAG Header

This header directly connects a JTAG interface cable to the JTAG pins to the on-board User FPGA JTAG chain. The pinout of this header matches the pinout of the Xilinx Platform Cable USB II. This allows the direct usage of Xilinx software-tools like Chipscope or iMPACT with the Platform Cable USB II. The connector is a 2 mm dual row shrouded header.

### 10.5.1 Connector Type

| | |
|---|---|
| **Pin-Count** | 14 |
| **Connector Type** | 2.00 mm Pitch Milli-Grid™ Header |
| **Source & Order Info** | Molex 877601416 or compatible |

### 10.5.2 Pin Assignment

| Pin | Signal | Description |
|---|---|---|
| 1 | NC | Not Connected |
| 2 | $V_{REF}$ | JTAG Reference Voltage (3.3V) |
| 3 | GND | Ground |
| 4 | TMS | Test Mode Select Input |
| 5 | GND | Ground |
| 6 | TCK | Test Clock |
| 7 | GND | Ground |
| 8 | TDO | Test Data Output (TAP Controller: TDI) |
| 9 | GND | Ground |
| 10 | TDI | Test Data Input (TAP Controller: TDO) |
| 11 | GND | not connected on the TXMC635 |
| 12 | TRST# | not connected on the TXMC635 |
| 13 | PGND | Used on TXMC635 for XILINX Header present detection |
| 14 | NC | HALT_INIT_WP signal. Optional. Not connected on the TXMC635 |

Table 10-3: Pin Assignment JTAG Header X2

# 10.6 X3 Debug-Connector

## 10.6.1 Connector Type

| | |
|---|---|
| **Pin-Count** | 20 |
| **Connector Type** | 20-pin, 1 mm FPC (Flexible Printed Circuit) Connector |
| **Source & Order Info** | AMP 2-487951-0 / 2-84953-0 or<br>Molex 0522072060 |

## 10.6.2 Pin Assignment

| Pin | Signal | I/O | Description |
|---|---|---|---|
| 1 | JTAG SEL | O | A 4.7k pull-up to 3.3 Volt is located on the TXMC635 |
| 2 | 3.3V | O | JTAG reference I/O voltage |
| 3 | TDO | O | Test Data Output (Input at JTAG Interface) |
| 4 | GND | - | Ground |
| 5 | TDI | I | Test Data Input (Output at JTAG Interface) |
| 6 | TMS | I | Test Mode Select Input |
| 7 | GND | - | Ground |
| 8 | TCK | I | Test Clock |
| 9 | GND | - | Ground |
| 10 | USER_RxD | I | FPGA UART Receive Data (Input) |
| 11 | 1.5V | O | UART reference I/O voltage |
| 12 | USER_TxD | O | FPGA UART Transmit Data (Output) |
| 13 | GND | - | Ground |
| 14 | FPGA_RxD | I | Used for Configuration FPGA on TXMC635 |
| 15 | 3.3V | O | 3.3V reference I/O voltage |
| 16 | FPGA_TxD | O | Used for Configuration FPGA on TXMC635 |
| 17 | GND | - | Ground |
| 18 | 3.3V | O | +3.3 Volt |
| 19 | 1.5V | O | User signal reference I/O voltage |
| 20 | GPIO_BUT | I | User signal connected to the FPGA, A 4.7k pull-up to 1.5 Volt is located on the TXMC635. |

Table 10-4: Pin Assignment Debug Connector X3

# 11 Appendix A

This appendix contains the signal to pin assignments for the Spartan6 FPGA.

```
## ################################################################################################################ ##
##             TEWS TECHNOLOGIES                ##
## ################################################################################################################ ##
##
## Project Name  : TXMC635 FPGA Example
## File Name       : top_entity.ucf
## Target Device : XC6SLX45T-2FGG484
## Design Tool   : Xilinx ISE Design Suit Embedded 14.7
## Simulation Tool : -
##
## Description   : Constraint file TXMC635-xx FPGA application. Refer to user manual for more
##       Information
##
## Owner     : TEWS TECHNOLOGIES GmbH
##       Am Bahnhof 7
##       D-25469 Halstenbek
##
##       Tel.: +49 / (0)4101 / 4058-0
##       Fax.: +49 / (0)4101 / 4058-19
##       e-mail: support@tews.com
##
##       Copyright (c) 2014
##       TEWS TECHNOLOGIES GmbH
##
## History   :
##   Version 1 : (NP, 21.04.2015)
##        Initial Version based on TXMC633 UCF File
##        Changes due to changed Front I/O Interface of the TXMC635.
##        - ADC Interface; DAC Interface; reduced TTL I/O Interface; IIC Interface
##
## Comments    : none
##
## ################################################################################################################ ##


## ################################################################################################################ ##
## Section: Miscellaneous
## ################################################################################################################ ##

# Set VCC aux power supply values (necessary for Spartan-6 architecture)
config vccaux = 3.3;

# Additional Bank Supply Information find below:
#
#  Bank No.   Supply
# ----------------------------
#   0    3.3V
#   1    1.5V
#   2    3.3V
#   3    3.3V
#

# Prohibit usage of pins that are not allowed for user I/O
config prohibit         = "Y20";    # Bank 2, CCLK

config prohibit         = "AA21";   # Bank 2, FPGA_M0
config prohibit         = "Y19";    # Bank 2, FPGA_M1

config prohibit         = "AB20";   # Bank 2, MOSI/MISO0/CSI_B
config prohibit         = "AA20";   # Bank 2, D0/MISO1
config prohibit         = "R13";    # Bank 2, D1/MISO2
config prohibit         = "T14";    # Bank 2, D2/MISO3
config prohibit         = "AA6";    # Bank 2, FPGA_D3
config prohibit         = "AB6";    # Bank 2, FPGA_D4
config prohibit         = "Y5";     # Bank 2, FPGA_D5
config prohibit         = "AB5";    # Bank 2, FPGA_D6
config prohibit         = "W9";     # Bank 2, FPGA_D7
```

```
config prohibit         = "Y8";     # Bank 2, FPGA_RDWR_B
config prohibit         = "Y4";     # Bank 2, FPGA_INIT_B
config prohibit         = "AA3";    # Bank 2, CSO_B

config prohibit         = "C3";     # Bank 0, HSWAPEN (pull-up resistor)


## ##############################################################################################
## Section: GTP Transceiver
## ##############################################################################################

# Location Constraints
net "PER2_P"      loc = "B6";  # Bank 101, MGT/PCI Express TX_P
net "PER2_N"      loc = "A6";  # Bank 101, MGT/PCI Express TX_N
net "PET2_P"      loc = "D7";  # Bank 101, MGT/PCI Express RX_P
net "PET2_N"      loc = "C7";  # Bank 101, MGT/PCI Express RX_N

config prohibit       = "B8";  # Bank 101, XMC P16 MGT TX_P Lane 1
config prohibit       = "A8";  # Bank 101, XMC P16 MGT TX_N Lane 1
config prohibit       = "D9";  # Bank 101, XMC P16 MGT RX_P Lane 1
config prohibit       = "C9";  # Bank 101, XMC P16 MGT RX_N Lane 1

net "PCIe_CLK125_P"     loc = "A10"; # Bank 101, PCI Express Reference Clock 125 MHz (CLK_P)
               #     via SI5338 (modified)
net "PCIe_CLK125_N"     loc = "B10"; # Bank 101, PCI Express Reference Clock 125 MHz (CLK_N)
               #     via SI5338 (modified)

config prohibit       = "B14"; # Bank 123, XMC P16 MGT TX_P Lane 2
config prohibit       = "A14"; # Bank 123, XMC P16 MGT TX_N Lane 2
config prohibit       = "D13"; # Bank 123, XMC P16 MGT RX_P Lane 2
config prohibit       = "C13"; # Bank 123, XMC P16 MGT RX_N Lane 2
config prohibit       = "B16"; # Bank 123, XMC P16 MGT RX_P Lane 3
config prohibit       = "A16"; # Bank 123, XMC P16 MGT RX_N Lane 3
config prohibit       = "D15"; # Bank 123, XMC P16 MGT RX_P Lane 3
config prohibit       = "C15"; # Bank 123, XMC P16 MGT RX_N Lane 3

config prohibit       = "A12"; # Bank 123, PCI Express Reference Clock 100 MHz (CLK_P)
               #     via PI7C9X2G404 (PCI Express Switch)
config prohibit       = "B12"; # Bank 123, PCI Express Reference Clock 100 MHz (CLK_N)
               #     via PI7C9X2G404 (PCI Express Switch)

# Additional Constraints
NET "*/gt_refclk_out(0)" TNM_NET = GT_REFCLK_OUT;
TIMESPEC TS_GT_REFCLK_OUT = PERIOD GT_REFCLK_OUT 8 ns HIGH 50 % ;


## ##############################################################################################
## Section: I/O Lines
## ##############################################################################################

# Define I/O Standard
net "FPGA_OE[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "FPGA_IO[*]"     iostandard = LVCMOS33;   # Bank 0,3


# Location Constraints
net "FPGA_OE[0]"        loc = "N1";     # Bank 3
net "FPGA_OE[1]"        loc = "M1";     # Bank 3
net "FPGA_OE[2]"        loc = "Y1";     # Bank 3
net "FPGA_OE[3]"        loc = "V2";     # Bank 3
net "FPGA_OE[4]"        loc = "Y2";     # Bank 3
net "FPGA_OE[5]"        loc = "Y3";     # Bank 3
net "FPGA_OE[6]"        loc = "U1";     # Bank 3
net "FPGA_OE[7]"        loc = "V3";     # Bank 3
net "FPGA_OE[8]"        loc = "U3";     # Bank 3
net "FPGA_OE[9]"        loc = "N3";     # Bank 3
net "FPGA_OE[10]"       loc = "N6";     # Bank 3
net "FPGA_OE[11]"       loc = "P8";     # Bank 3
net "FPGA_OE[12]"       loc = "R7";     # Bank 3
net "FPGA_OE[13]"       loc = "M2";     # Bank 3
net "FPGA_OE[14]"       loc = "G1";     # Bank 3
net "FPGA_OE[15]"       loc = "D3";     # Bank 0
```

```
net "FPGA_OE[16]"       loc = "R1";     # Bank 3
net "FPGA_OE[17]"       loc = "T3";     # Bank 3
net "FPGA_OE[18]"       loc = "P5";     # Bank 3
net "FPGA_OE[19]"       loc = "U4";     # Bank 3
net "FPGA_OE[20]"       loc = "H4";     # Bank 3
net "FPGA_OE[21]"       loc = "J7";     # Bank 3
net "FPGA_OE[22]"       loc = "P4";     # Bank 3
net "FPGA_OE[23]"       loc = "T6";     # Bank 3
net "FPGA_OE[24]"       loc = "N7";     # Bank 3
net "FPGA_OE[25]"       loc = "R4";     # Bank 3
net "FPGA_OE[26]"       loc = "G13";    # Bank 0
net "FPGA_OE[27]"       loc = "F17";    # Bank 0
net "FPGA_OE[28]"       loc = "H1";     # Bank 3
net "FPGA_OE[29]"       loc = "E1";     # Bank 3
net "FPGA_OE[30]"       loc = "D2";     # Bank 3
net "FPGA_OE[31]"       loc = "B3";     # Bank 0
net "FPGA_OE[32]"       loc = "M6";     # Bank 3
net "FPGA_OE[33]"       loc = "H3";     # Bank 3
net "FPGA_OE[34]"       loc = "C4";     # Bank 0
net "FPGA_OE[35]"       loc = "G4";     # Bank 3
net "FPGA_OE[36]"       loc = "G3";     # Bank 3
net "FPGA_OE[37]"       loc = "K3";     # Bank 3
net "FPGA_OE[38]"       loc = "J4";     # Bank 3
net "FPGA_OE[39]"       loc = "T4";     # Bank 3
net "FPGA_OE[40]"       loc = "D4";     # Bank 0
net "FPGA_OE[41]"       loc = "K4";     # Bank 3
net "FPGA_OE[42]"       loc = "F7";     # Bank 0
net "FPGA_OE[43]"       loc = "F9";     # Bank 0
net "FPGA_OE[44]"       loc = "E5";     # Bank 0
net "FPGA_OE[45]"       loc = "E6";     # Bank 0
net "FPGA_OE[46]"       loc = "J6";     # Bank 3
net "FPGA_OE[47]"       loc = "G8";     # Bank 0

net "FPGA_IO[0]"        loc = "P1";     # Bank 3, Front I/O X1
net "FPGA_IO[1]"        loc = "P6";     # Bank 3, Front I/O X1
net "FPGA_IO[2]"        loc = "AA2";    # Bank 3, Front I/O X1
net "FPGA_IO[3]"        loc = "W1";     # Bank 3, Front I/O X1
net "FPGA_IO[4]"        loc = "AA1";    # Bank 3, Front I/O X1
net "FPGA_IO[5]"        loc = "V1";     # Bank 3, Front I/O X1
net "FPGA_IO[6]"        loc = "W3";     # Bank 3, Front I/O X1
net "FPGA_IO[7]"        loc = "W4";     # Bank 3, Front I/O X1
net "FPGA_IO[8]"        loc = "T1";     # Bank 3, Front I/O X1
net "FPGA_IO[9]"        loc = "P2";     # Bank 3, Front I/O X1
net "FPGA_IO[10]"       loc = "F2";     # Bank 3, Front I/O X1
net "FPGA_IO[11]"       loc = "H2";     # Bank 3, Front I/O X1
net "FPGA_IO[12]"       loc = "K1";     # Bank 3, Front I/O X1
net "FPGA_IO[13]"       loc = "J1";     # Bank 3, Front I/O X1
net "FPGA_IO[14]"       loc = "F1";     # Bank 3, Front I/O X1
net "FPGA_IO[15]"       loc = "B1";     # Bank 3, Front I/O X1
net "FPGA_IO[16]"       loc = "T2";     # Bank 3, Front I/O X1
net "FPGA_IO[17]"       loc = "R3";     # Bank 3, Front I/O X1
net "FPGA_IO[18]"       loc = "T5";     # Bank 3, Front I/O X1
net "FPGA_IO[19]"       loc = "V5";     # Bank 3, Front I/O X1
net "FPGA_IO[20]"       loc = "L4";     # Bank 3, Front I/O X1
net "FPGA_IO[21]"       loc = "M3";     # Bank 3, Front I/O X1
net "FPGA_IO[22]"       loc = "M4";     # Bank 3, Front I/O X1
net "FPGA_IO[23]"       loc = "A3";     # Bank 0, Front I/O X1
net "FPGA_IO[24]"       loc = "C1";     # Bank 3, Front I/O X1
net "FPGA_IO[25]"       loc = "B2";     # Bank 0, Front I/O X1
net "FPGA_IO[26]"       loc = "D17";    # Bank 0, Front I/O X1
net "FPGA_IO[27]"       loc = "D18";    # Bank 0, Front I/O X1
net "FPGA_IO[28]"       loc = "K2";     # Bank 3, Front I/O X1
net "FPGA_IO[29]"       loc = "D1";     # Bank 3, Front I/O X1
net "FPGA_IO[30]"       loc = "A2";     # Bank 0, Front I/O X1
net "FPGA_IO[31]"       loc = "F3";     # Bank 3, Front I/O X1
net "FPGA_IO[32]"       loc = "E3";     # Bank 3, Front I/O X1
net "FPGA_IO[33]"       loc = "J3";     # Bank 3, Front I/O X1
net "FPGA_IO[34]"       loc = "E4";     # Bank 3, Front I/O X1
net "FPGA_IO[35]"       loc = "D5";     # Bank 0, Front I/O X1
net "FPGA_IO[36]"       loc = "A4";     # Bank 0, Front I/O X1
net "FPGA_IO[37]"       loc = "A5";     # Bank 0, Front I/O X1
net "FPGA_IO[38]"       loc = "F5";     # Bank 3, Front I/O X1
net "FPGA_IO[39]"       loc = "H6";     # Bank 3, Front I/O X1
```

```
net "FPGA_IO[40]"       loc = "C5";     # Bank 0, Front I/O X1
net "FPGA_IO[41]"       loc = "K5";     # Bank 3, Front I/O X1
net "FPGA_IO[42]"       loc = "F8";     # Bank 0, Front I/O X1
net "FPGA_IO[43]"       loc = "L3";     # Bank 3, Front I/O X1
net "FPGA_IO[44]"       loc = "H5";     # Bank 3, Front I/O X1
net "FPGA_IO[45]"       loc = "G6";     # Bank 3, Front I/O X1
net "FPGA_IO[46]"       loc = "G7";     # Bank 3, Front I/O X1
net "FPGA_IO[47]"       loc = "K7";     # Bank 3, Front I/O X1

config prohibit         = "AA4";    # Bank 2, PMC Back I/O P14
config prohibit         = "AB4";    # Bank 2, PMC Back I/O P14
config prohibit         = "W6";     # Bank 2, PMC Back I/O P14
config prohibit         = "Y6";     # Bank 2, PMC Back I/O P14
config prohibit         = "T7";     # Bank 2, PMC Back I/O P14
config prohibit         = "U6";     # Bank 2, PMC Back I/O P14
config prohibit         = "Y7";     # Bank 2, PMC Back I/O P14
config prohibit         = "AB7";    # Bank 2, PMC Back I/O P14
config prohibit         = "V7";     # Bank 2, PMC Back I/O P14
config prohibit         = "W8";     # Bank 2, PMC Back I/O P14
config prohibit         = "AA8";    # Bank 2, PMC Back I/O P14
config prohibit         = "AB8";    # Bank 2, PMC Back I/O P14
config prohibit         = "T8";     # Bank 2, PMC Back I/O P14
config prohibit         = "U8";     # Bank 2, PMC Back I/O P14
config prohibit         = "R9";     # Bank 2, PMC Back I/O P14
config prohibit         = "R8";     # Bank 2, PMC Back I/O P14
config prohibit         = "Y9";     # Bank 2, PMC Back I/O P14
config prohibit         = "AB9";    # Bank 2, PMC Back I/O P14
config prohibit         = "U9";     # Bank 2, PMC Back I/O P14
config prohibit         = "V9";     # Bank 2, PMC Back I/O P14
config prohibit         = "AA10";   # Bank 2, PMC Back I/O P14
config prohibit         = "AB10";   # Bank 2, PMC Back I/O P14
config prohibit         = "W10";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y10";    # Bank 2, PMC Back I/O P14
config prohibit         = "T10";    # Bank 2, PMC Back I/O P14
config prohibit         = "U10";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y11";    # Bank 2, PMC Back I/O P14
config prohibit         = "AB11";   # Bank 2, PMC Back I/O P14
config prohibit         = "V11";    # Bank 2, PMC Back I/O P14
config prohibit         = "W11";    # Bank 2, PMC Back I/O P14
config prohibit         = "R11";    # Bank 2, PMC Back I/O P14
config prohibit         = "T11";    # Bank 2, PMC Back I/O P14
config prohibit         = "AA12";   # Bank 2, PMC Back I/O P14
config prohibit         = "AB12";   # Bank 2, PMC Back I/O P14
config prohibit         = "W12";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y12";    # Bank 2, PMC Back I/O P14
config prohibit         = "T12";    # Bank 2, PMC Back I/O P14
config prohibit         = "U12";    # Bank 2, PMC Back I/O P14
config prohibit         = "V13";    # Bank 2, PMC Back I/O P14
config prohibit         = "W13";    # Bank 2, PMC Back I/O P14
config prohibit         = "AA14";   # Bank 2, PMC Back I/O P14
config prohibit         = "AB14";   # Bank 2, PMC Back I/O P14
config prohibit         = "W14";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y14";    # Bank 2, PMC Back I/O P14
config prohibit         = "U14";    # Bank 2, PMC Back I/O P14
config prohibit         = "U13";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y15";    # Bank 2, PMC Back I/O P14
config prohibit         = "AB15";   # Bank 2, PMC Back I/O P14
config prohibit         = "AA16";   # Bank 2, PMC Back I/O P14
config prohibit         = "AB16";   # Bank 2, PMC Back I/O P14
config prohibit         = "Y16";    # Bank 2, PMC Back I/O P14
config prohibit         = "W15";    # Bank 2, PMC Back I/O P14
config prohibit         = "U16";    # Bank 2, PMC Back I/O P14
config prohibit         = "V15";    # Bank 2, PMC Back I/O P14
config prohibit         = "T15";    # Bank 2, PMC Back I/O P14
config prohibit         = "U15";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y17";    # Bank 2, PMC Back I/O P14
config prohibit         = "AB17";   # Bank 2, PMC Back I/O P14
config prohibit         = "AA18";   # Bank 2, PMC Back I/O P14
config prohibit         = "AB18";   # Bank 2, PMC Back I/O P14
config prohibit         = "W17";    # Bank 2, PMC Back I/O P14
config prohibit         = "Y18";    # Bank 2, PMC Back I/O P14
config prohibit         = "V17";    # Bank 2, PMC Back I/O P14
config prohibit         = "W18";    # Bank 2, PMC Back I/O P14
```

```
## ############################################################################################################ ##
## Section: ADC Interface
## ############################################################################################################ ##

net "CNV_ADC[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "CS_ADC_n[*]"      iostandard = LVCMOS33;   # Bank 0,3
net "SCK_ADC[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "DIN_ADC[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "SDO_ADC[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "RESET_ADC"      iostandard = LVCMOS33;   # Bank 0,3
net "PD_ADC"       iostandard = LVCMOS33;   # Bank 0,3
net "BUSY_ADC[*]"      iostandard = LVCMOS15;   # Bank 1

net "CNV_ADC[0]"     loc = "A19";       # Bank 0,3
net "CNV_ADC[1]"     loc = "G9";      # Bank 0,3
net "CNV_ADC[2]"     loc = "K6";      # Bank 0,3
net "CNV_ADC[3]"     loc = "L1";      # Bank 0,3
net "CS_ADC_n[0]"      loc = "D19";       # Bank 0,3
net "CS_ADC_n[1]"      loc = "C18";       # Bank 0,3
net "CS_ADC_n[2]"      loc = "P7";      # Bank 0,3
net "CS_ADC_n[3]"      loc = "K8";      # Bank 0,3
net "SCK_ADC[0]"     loc = "B18";       # Bank 0,3
net "SCK_ADC[1]"     loc = "G11";       # Bank 0,3
net "SCK_ADC[2]"     loc = "M7";      # Bank 0,3
net "SCK_ADC[3]"     loc = "H8";      # Bank 0,3
net "DIN_ADC[0]"     loc = "A18";       # Bank 0,3
net "DIN_ADC[1]"     loc = "H12";       # Bank 0,3
net "DIN_ADC[2]"     loc = "M8";      # Bank 0,3
net "DIN_ADC[3]"     loc = "N4";      # Bank 0,3
net "SDO_ADC[0]"     loc = "C19";       # Bank 0,3
net "SDO_ADC[1]"     loc = "F10";       # Bank 0,3
net "SDO_ADC[2]"     loc = "L6";      # Bank 0,3
net "SDO_ADC[3]"     loc = "P3";      # Bank 0,3
net "BUSY_ADC[0]"      loc = "R15";       # Bank 1
net "BUSY_ADC[1]"      loc = "R16";       # Bank 1
net "BUSY_ADC[2]"      loc = "V19";       # Bank 1
net "BUSY_ADC[3]"      loc = "P17";       # Bank 1
net "RESET_ADC"      loc = "A20";       # Bank 0,3
net "PD_ADC"       loc = "B20";       # Bank 0,3

## ############################################################################################################ ##
## Section: DAC Interface
## ############################################################################################################ ##

## ############################################################################################################ ##
## Section:  DAC Interface
## ############################################################################################################ ##

net "SYNC_DAC_n[*]"    slow | drive = 8;
net "LDAC_DAC_n[*]"    slow | drive = 8;
net "SCLK_DAC[*]"      slow | drive = 8;
net "SDIN_DAC[*]"      slow | drive = 8;
net "SDO_DAC[*]"     slow | drive = 8;
net "RSTIN_DAC_n"      slow | drive = 8;
net "CLR_DAC_n"      slow | drive = 8;

net "SYNC_DAC_n[*]"    iostandard = LVCMOS33;   # Bank 0,3
net "LDAC_DAC_n[*]"    iostandard = LVCMOS33;   # Bank 0,3
net "SCLK_DAC[*]"      iostandard = LVCMOS33;   # Bank 0,3
net "SDIN_DAC[*]"      iostandard = LVCMOS33;   # Bank 0,3
net "SDO_DAC[*]"     iostandard = LVCMOS33;   # Bank 0,3
net "RSTIN_DAC_n"      iostandard = LVCMOS33;   # Bank 0,3
net "CLR_DAC_n"      iostandard = LVCMOS33;   # Bank 0,3

net "SYNC_DAC_n[0]"    loc = "F15";       # Bank 0,3
net "SYNC_DAC_n[1]"    loc = "A17";       # Bank 0,3
net "LDAC_DAC_n[0]"    loc = "F14";       # Bank 0,3
net "LDAC_DAC_n[1]"    loc = "G16";       # Bank 0,3
net "SCLK_DAC[0]"      loc = "H10";       # Bank 0,3
net "SCLK_DAC[1]"      loc = "H14";       # Bank 0,3
net "SDIN_DAC[0]"      loc = "F16";       # Bank 0,3
net "SDIN_DAC[1]"      loc = "G15";       # Bank 0,3
net "SDO_DAC[0]"     loc = "E16";       # Bank 0,3
```

```
net "SDO_DAC[1]"     loc = "C17";       # Bank 0,3
net "RSTIN_DAC_n"      loc = "H11";       # Bank 0,3
net "CLR_DAC_n"      loc = "H13";       # Bank 0,3


## ############################################################################################## ##
## Section: DDR3 Memory (MCB1)
## ############################################################################################## ##

# MCB 3, I/O Termination
net "DDR_DQ[*]"       in_term = none;
net "DDR_?DQS_?"      in_term = none;

# MCB 3, I/O Standards
net "DDR_DQ[*]"      iostandard = SSTL15_II;  # 1.5V
net "DDR_A[*]"       iostandard = SSTL15_II;  # 1.5V
net "DDR_BA[*]"      iostandard = SSTL15_II;  # 1.5V
net "DDR_?DQS_?"     iostandard = DIFF_SSTL15_II; # 1.5V
net "DDR_CK_?"       iostandard = DIFF_SSTL15_II; # 1.5V
net "DDR_CKe"      iostandard = SSTL15_II;  # 1.5V
net "DDR_RAS_n"      iostandard = SSTL15_II;  # 1.5V
net "DDR_CAS_n"      iostandard = SSTL15_II;  # 1.5V
net "DDR_WE_n"       iostandard = SSTL15_II;  # 1.5V
net "DDR_ODT"      iostandard = SSTL15_II;  # 1.5V
net "DDR_RESET_n"      iostandard = LVCMOS15;   # 1.5V
net "DDR_?DM"      iostandard = SSTL15_II;  # 1.5V
net "DDR_RZQ"      iostandard = SSTL15_II;  # 1.5V
net "DDR_ZIO"      iostandard = SSTL15_II;  # 1.5V


# MCB 3, Pin Location Constraints for Clock, Masks, Address, and Controls
net "DDR_A[0]"        loc = "H21";    # Bank 1
net "DDR_A[1]"        loc = "H22";    # Bank 1
net "DDR_A[2]"        loc = "G22";    # Bank 1
net "DDR_A[3]"        loc = "J20";    # Bank 1
net "DDR_A[4]"        loc = "H20";    # Bank 1
net "DDR_A[5]"        loc = "M20";    # Bank 1
net "DDR_A[6]"        loc = "M19";    # Bank 1
net "DDR_A[7]"        loc = "G20";    # Bank 1
net "DDR_A[8]"        loc = "E20";    # Bank 1
net "DDR_A[9]"        loc = "E22";    # Bank 1
net "DDR_A[10]"         loc = "J19";    # Bank 1
net "DDR_A[11]"         loc = "H19";    # Bank 1
net "DDR_A[12]"         loc = "F22";    # Bank 1
config prohibit         = "G19";    # Bank 1, DDR_A[13]
config prohibit         = "F20";    # Bank 1, DDR_A[14]

net "DDR_BA[0]"         loc = "K17";    # Bank 1
net "DDR_BA[1]"         loc = "L17";    # Bank 1
net "DDR_BA[2]"         loc = "K18";    # Bank 1

net "DDR_CK_P"        loc = "K20";    # Bank 1
net "DDR_CK_N"        loc = "L19";    # Bank 1

net "DDR_DQ[0]"         loc = "R20";    # Bank 1
net "DDR_DQ[1]"         loc = "R22";    # Bank 1
net "DDR_DQ[2]"         loc = "P21";    # Bank 1
net "DDR_DQ[3]"         loc = "P22";    # Bank 1
net "DDR_DQ[4]"         loc = "L20";    # Bank 1
net "DDR_DQ[5]"         loc = "L22";    # Bank 1
net "DDR_DQ[6]"         loc = "M21";    # Bank 1
net "DDR_DQ[7]"         loc = "M22";    # Bank 1
net "DDR_DQ[8]"         loc = "T21";    # Bank 1
net "DDR_DQ[9]"         loc = "T22";    # Bank 1
net "DDR_DQ[10]"        loc = "U20";    # Bank 1
net "DDR_DQ[11]"        loc = "U22";    # Bank 1
net "DDR_DQ[12]"        loc = "W20";    # Bank 1
net "DDR_DQ[13]"        loc = "W22";    # Bank 1
net "DDR_DQ[14]"        loc = "Y21";    # Bank 1
net "DDR_DQ[15]"        loc = "Y22";    # Bank 1

net "DDR_CKE"         loc = "F21";    # Bank 1
```

```
net "DDR_ODT"          loc = "J22";    # Bank 1

net "DDR_LDQS_P"         loc = "N20";    # Bank 1
net "DDR_LDQS_N"         loc = "N22";    # Bank 1
net "DDR_UDQS_P"         loc = "V21";    # Bank 1
net "DDR_UDQS_N"         loc = "V22";    # Bank 1

net "DDR_CAS_n"          loc = "K22";    # Bank 1
net "DDR_RAS_n"          loc = "K21";    # Bank 1
net "DDR_WE_n"         loc = "K19";    # Bank 1

net "DDR_LDM"          loc = "N19";    # Bank 1
net "DDR_UDM"          loc = "P20";    # Bank 1

net "DDR_RESET_n"        loc = "H18";    # Bank 1

net "DDR_RZQ"          loc = "F18";    # Bank 1
net "DDR_ZIO"          loc = "P19";    # Bank 1

config prohibit          = "F19";    # Bank 1, DDR3 Reference Voltage
config prohibit          = "D22";    # Bank 1, DDR3 Reference Voltage
config prohibit          = "R19";    # Bank 1, DDR3 Reference Voltage


# Additional Constraints
config mcb_performance         = standard;   # General MCB constraints


## ##############################################################################################################
## Section: Clocking
## ##############################################################################################################

# I/O Standards
net "MCB_CLK"      iostandard = LVCMOS33;   # Bank 2


# Location Constraints
config prohibit          = "AB13";   # Bank 2

net "MCB_CLK"          loc = "Y13";    # Bank 2

config prohibit          = "M5";     # Bank 3


# Additional Constraints
net "MCB_CLK"       tnm_net = "MCB_CLK";
timespec "TS_MCB_CLK"       = period "MCB_CLK" 62.5 MHz high 50 %;


## ##############################################################################################################
## Section: Module Management
## ##############################################################################################################

# I/O Standards
net "DWNRST2_n"      iostandard = LVCMOS15;   # Bank 1
net "PULL_IN[*]"     iostandard = LVCMOS15;   # Bank 1


# Location Constraints
config prohibit          = "B22";    # Bank 1
config prohibit          = "J16";    # Bank 1
config prohibit          = "J17";    # Bank 1
config prohibit          = "C20";    # Bank 1
config prohibit          = "C22";    # Bank 1
config prohibit          = "L15";    # Bank 1
config prohibit          = "K16";    # Bank 1
config prohibit          = "D21";    # Bank 1

net "PULL_IN[0]"         loc = "U19";    # Bank 1, Group [ 0 : 15]
net "PULL_IN[1]"         loc = "T20";    # Bank 1, Group [ 0 : 15]
net "PULL_IN[2]"         loc = "N16";    # Bank 1, Group [16 : 31]
net "PULL_IN[3]"         loc = "P16";    # Bank 1, Group [16 : 31]
net "PULL_IN[4]"         loc = "M17";    # Bank 1, Group [32 : 47]
```

```
net "PULL_IN[5]"        loc = "M18";    # Bank 1, Group [32 : 47]

net "DWNRST2_n"         loc = "AB19";   # Bank 2, PCI Express Reset


## ############################################################################################## ##
## Section: General Purpose I/O
## ############################################################################################## ##

# I/O Standards
net "USER_LED[?]"      iostandard = LVCMOS15;   # Bank 1
net "SDA"        iostandard = LVCMOS15;   # Bank 1
net "SCL"        iostandard = LVCMOS15;   # Bank 1

# Location Constraints
net "USER_LED[0]"       loc = "M16";    # Bank 1
net "USER_LED[1]"       loc = "N15";    # Bank 1
net "SDA"           loc = "P18";    # Bank 1
net "SCL"           loc = "R17";    # Bank 1


config prohibit         = "T17";    # Bank 1, Debug Connector X3
config prohibit         = "T19";    # Bank 1, Debug Connector X3
config prohibit         = "T18";    # Bank 1, Debug Connector X3
```