The Embedded I/O Company

TEWS TECHNOLOGIES

TXMC633

Reconfigurable FPGA with 64 TTL I/O / 32 Differential I/O Lines

Version 1.0

User Manual

Issue 1.0.4

July 2018

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

TXMC633-10R

64 TTL Front I/O and 64 direct FPGA I/O Lines, XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-11R

32 Differential Front I/O and 64 direct FPGA I/O Lines, XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-12R

32 TTL and 16 Differential Front I/O and 64 direct FPGA I/O Lines, XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-13R

32 Differential M-LVDS Front I/O and 64 direct FPGA I/O Lines, XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-14R

32 TTL and 16 Differential M-LVDS Front I/O and 64 direct FPGA I/O Lines, XC6SLX45T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-20R

64 TTL Front I/O and 64 direct FPGA I/O Lines, XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-21R

32 Differential Front I/O and 64 direct FPGA I/O Lines, XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-22R

32 TTL and 16 Differential Front I/O and 64 direct FPGA I/O Lines, XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-23R

32 Differential M-LVDS Front I/O and 64 direct FPGA I/O Lines, XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3

TXMC633-24R

32 TTL and 16 Differential M-LVDS Front I/O and 64 direct FPGA I/O Lines, XC6SLX100T-2 Spartan-6 FPGA,128 MB DDR3

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

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

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

Style Conventions

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

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

Access terms are described as:

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

Issue	Description	Date
1.0.0	Initial issue	December 2014
1.0.1	Changes in the BCC firmware and also changes in the process description.	April 2015
1.0.2	Description of I/O Interface extended	July 2015
1.0.3	Alternative configurations SPI-Flash assembly.	May 2017
1.0.4	Supplementary description for clock generation.	July 2018

Table of Contents

List of Figures

List of Tables

1 Product Description

The TXMC633 is a standard single-width Switched Mezzanine Card (XMC) compatible module providing a user configurable XC6SLX45T-2 or XC6SLX100T-2 Spartan-6 FPGA.

The TXMC633-x0 has 64 ESD-protected TTL lines; the TXMC633-x1 provides 32 differential I/O lines using EIA 422 / EIA 485 compatible, ESD-protected line transceivers. The TXMC633-x2 provides 32 TTL and 16 differential I/Os. The TXMC633-x3 provides 32 differential I/O lines using Multipoint-LVDS Transceiver. The TXMC633-x4 provides 32 TTL and 16 differential I/O Multipoint-LVDS Transceiver.

For customer specific I/O extension or inter-board communication, the TXMC633-xx provides 64 FPGA I/Os 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.

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 tri-stated output. Each TTL I/O line has a pull-resistor. The pull-voltage level is programmable to be either +3.3V, +5V and additionally GND. The differential RS485 I/O lines are terminated by 120Ω resistors and the differential MLVDS I/O lines are terminated by 100Ω resistors.

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”).

User applications for the TXMC633 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 TXMC633. 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.

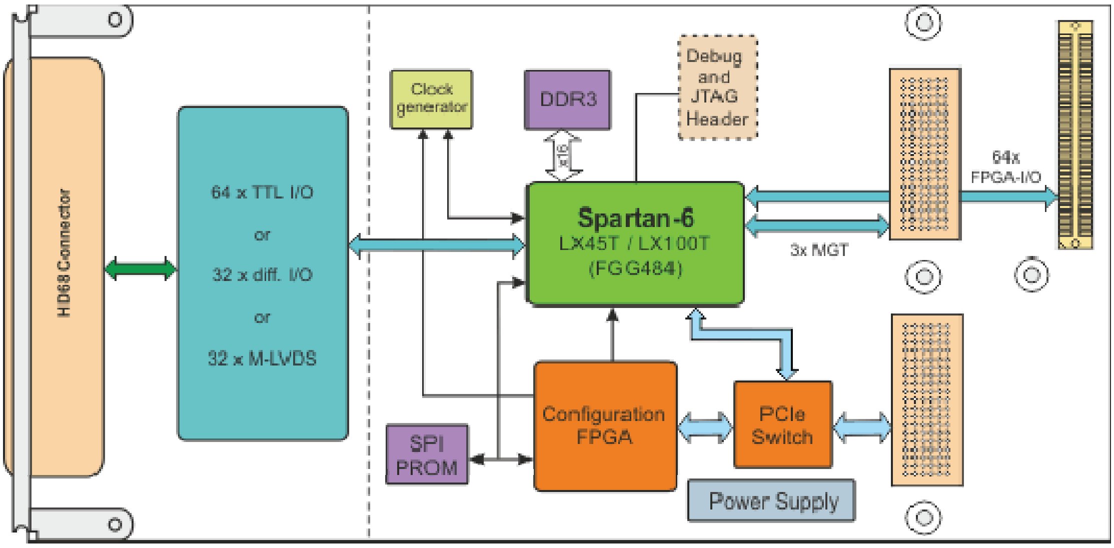


Figure 1-1 : Block Diagram

2 Technical Specification

<table>
<tr><td colspan="2"><b>XMC Interface</b></td></tr>
<tr><td><b>Mechanical Interface</b></td><td>Switched Mezzanine Card (XMC) Interface confirming to ANSI/VITA 42.0-2008 (Auxiliary Standard)
Standard single-width (149mm x 74mm)</td></tr>
<tr><td><b>Electrical Interface</b></td><td>PCI Express x1 Link (Base Specification 1.1) compliant interface conforming to ANSI/VITA 42.3-2006 (XMC PCI Express Protocol Layer Standard)</td></tr>
<tr><td colspan="2"><b>On-Board Devices</b></td></tr>
<tr><td><b>PCI Express Switch</b></td><td>PI7C9X2G404 (Pericom)</td></tr>
<tr><td><b>PCI Express to PCI Bridge</b></td><td>XIO2001 (Texas Instruments)</td></tr>
<tr><td><b>PCI Express Endpoint</b></td><td>Spartan-6 PCI Express Endpoint Block</td></tr>
<tr><td><b>User configurable FPGA</b></td><td>TXMC633-1x: XC6SLX45T-2 (Xilinx)
TXMC633-2x: XC6SLX100T-2 (Xilinx)</td></tr>
<tr><td><b>SPI-Flash</b></td><td>W25Q32FV (Winbond) or W25Q32JV (Winbond) 32 Mbit (contains TXMC633 FPGA Example)</td></tr>
<tr><td><b>DDR3 RAM</b></td><td>MT41J64M16 (Micron) or MT41K64M16 (Micron) 64 Meg x 16 Bit</td></tr>
<tr><td><b>Board Configuration FPGA</b></td><td>LCMXO2-2000HC (Lattice)</td></tr>
<tr><td colspan="2"><b>I/O Interface</b></td></tr>
<tr><td><b>I/O Channels</b></td><td>TXMC633-x0: 64 ESD-protected TTL lines
TXMC633-x1: 32 differential I/O lines
TXMC633-x2: 32 TTL and 16 differential I/O lines
TXMC633-x3: 32 M-LVDS I/O lines
TXMC633-x4: 32 TTL and 16 M-LVDS I/O lines.</td></tr>
<tr><td><b>I/O Transceiver</b></td><td>TXMC633-x0R: 74LVT126 (or compatible)
TXMC633-x1R: MAX3078E (or compatible)
TXMC633-x2R: 74LVT126 & MAX3078E (or compatible)
TXMC633-x3R: SN65MLVD206 (or compatible)
TXMC633-x4R: 74LVT126 & SN65MLVD206 (or compatible)</td></tr>
<tr><td><b>I/O Connector</b></td><td>Front I/O HD68 SCSI-3 type Connector
(AMP 787082-7 or compatible)
PMC P14 I/O (64 pin Mezzanine Connector)
XMC P16 I/O (114 pin Mezzanine Connector)</td></tr>
<tr><td colspan="2"><b>Physical Data</b></td></tr>
<tr><td><b>Power Requirements</b></td><td>Depends on FPGA design
With TXMC633 FPGA Example Design / without external load
<table>
<tr><td></td><td>typical @ +5V VPWR</td><td>typical @ +12V VPWR</td></tr>
<tr><td>TXMC633-xx</td><td>0.650 A</td><td>0.300 A</td></tr>
</table></td></tr>
<tr><td><b>Temperature Range</b></td><td>
<table>
<tr><td>Operating</td><td>-40°C to +85 °C</td></tr>
<tr><td>Storage</td><td>-40°C to +85°C</td></tr>
</table></td></tr>
</table>

MTBF	TXMC633-xx: 320000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G_B 20°C. The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation.
Humidity	5 – 95 % non-condensing
Weight	TXMC633-xx: 130g

Table 2-1 : Technical Specification

3 Handling and Operation Instruction

3.1 ESD Protection

The TXMC633 is sensitive to static electricity. Packing, unpacking and all other handling of the TXMC633 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 TXMC633 from carrier board please keep the mechanical stress as low as possible.

4 PCI Device Topology

The TXMC633 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 TXMC633 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.

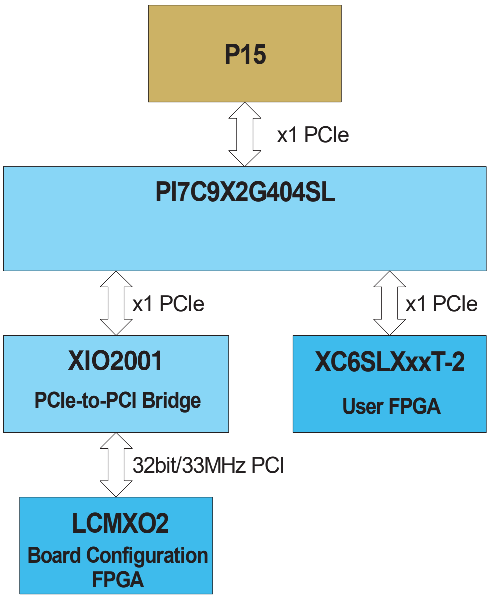


Figure 4-1 : PCIe/PCI Device Topology

Device	Vendor ID	Device ID	Class Code	Description (as shown by lspci)
PI7C9X2G404SL	0x12D8 (Pericom)	0x2404	0x060400	PCI bridge: 0x04h to indicate device as PCI-to-PCI Bridge 0x06h to indicate device as Bridge device
XIO2001	0x104C (Texas Instruments)	0x8240	0x060400	PCI bridge: Texas Instruments 0x04h to indicate device as PCI-to-PCI Bridge 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)	0x9279	0x068000	Bridge Device: TEWS Technologies GmbH Device 9279 (TXMC633).

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	9279 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	9279 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	TXMC633 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 0: Spartan6 User FPGA controls Pull Resistor 1: MachXO2 FPGA controls Pull Resistor	R/W	0
4	S6_LINK_ENA	1: Spartan6 to PCIe-Switch LINK is enabled 0: Spartan6 to PCIe-Switch LINK is disabled	R/W	1
3	FP_INIT_STAT	User FPGA INIT_B Pin Status 0: FPGA INIT_B Pin Level is Low (active) 1: FPGA INIT_B Pin Level is High (not active)	R	x
2	FP_DONE_STAT	User FPGA DONE Pin Status The FPGA Done pin is high in case of successful FPGA configuration. 0: FPGA DONE Pin Level is Low (not active) 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. User FPGA Re-Configuration 1: Set all FPGA I/O pins to High-Z and prepare a User FPGA Re-Configuration 1 → 0: Start User FPGA Re-Configuration	R/W	0
0	FP_CFG_MD	Set User FPGA Configuration Mode 0: Master Serial / SPI 1: Slave SelectMap (Parallel) 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 Write Data Register for direct SelectMap FPGA programming mode 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 Supported Instructions: 0x02 – Page Program 0x20 – Sector Erase 0x60 – Chip Erase 0x03 – Read Data 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 0: Disable ISP Mode 1: Enable ISP Mode 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. When set, the MachXO2 CPLD is both SPI Flash Master and FPGA Configuration Interface Master. Must be set to 1 for direct SelectMap FPGA or SPI Flash programming. 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. 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 Writing a '1' sets the Instruction Busy Bit in the ISP Status Register (if not already set). Breaks any ISP SPI instruction in progress and resets the ISP SPI logic. Check the Instruction Busy Bit in the ISP Status Register for reset done status. Always read as '0'.	R/W	0
0	ISP_SPI_INS_CMD	ISP SPI Start Instruction Command Bit Writing a '1' sets the SPI Instruction Busy Bit in the ISP Status Register and starts the configured SPI instruction. Ignored (lost) while the Instruction Busy Bit is set in the ISP Status Register. 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 Set & Cleared automatically by HW. Includes SPI Flash internal program/erase times. When clear again after being set, a new ISP SPI instruction may be started. Capable of generating an event based interrupt. 0: No ISP SPI Instruction in Progress 1: ISP SPI Instruction in Progress	R	0
0	ISP_SPI_ DAT_BSY	ISP SPI Data Transfer Busy Status Set & Cleared automatically by HW. Does not include SPI Flash internal program/erase times. 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). Capable of generating an event based interrupt. 0: No ISP SPI Data Transfer in Progress 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 0: Interrupt Disabled 1: Interrupt Enabled While disabled, the corresponding bit in the Interrupt Status Register is ‘0’. Disabling interrupts does not affect the interrupt source.	R/W	0
0	ISP_DAT_IE	ISP SPI Page Data Request Event Interrupt Enable 0: Interrupt Disabled 1: Interrupt Enabled While disabled, the corresponding bit in the Interrupt Status Register is ‘0’. 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 When set, the PCI INTA# interrupt is asserted. The Interrupt is cleared by writing a ‘1’. 0: Interrupt not active or disabled 1: Interrupt active and enabled	R/C	0
0	ISP_DAT_IS	ISP SPI Page Data Done Event Interrupt Status When set, the PCI INTA# interrupt is asserted. The Interrupt is cleared by writing a ‘1’. 0: Interrupt not active or disabled 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 Value could be changed only if PULL_CNT is set to MachXO2 controlling. 11 : pull-down 10 : pull-up to 3.3V 01 : pull-up to 5V 00 : No pull-up or pull-down I/O lines are summarized in the following groups. G0 = I/O_0 … I/O_15 G1 = I/O_16 … I/O_31 G2 = I/O_32 … I/O_47 G3 = I/O_48 … I/O_63	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 64 I/O Lines are divided into four 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 TXMC633 Serial Number - 0xF8

Bit	Symbol	Description	Access	Reset Value
31:0	S_NUMBER	The value is the unique serial number of each TXMC633 module	r	-

Table 5-10 : TXMC633 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 TXMC633 module.	r	-

Table 5-11 : MachXO2 - FPGA Code Version

Example:

0x0000_0100 => bit 32 downto 16 : reserved

0x0000_0100 => bit 15 downto 8 : Major FPGA Code Version

0x0000_0100 => bit 7 downto 0 : 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

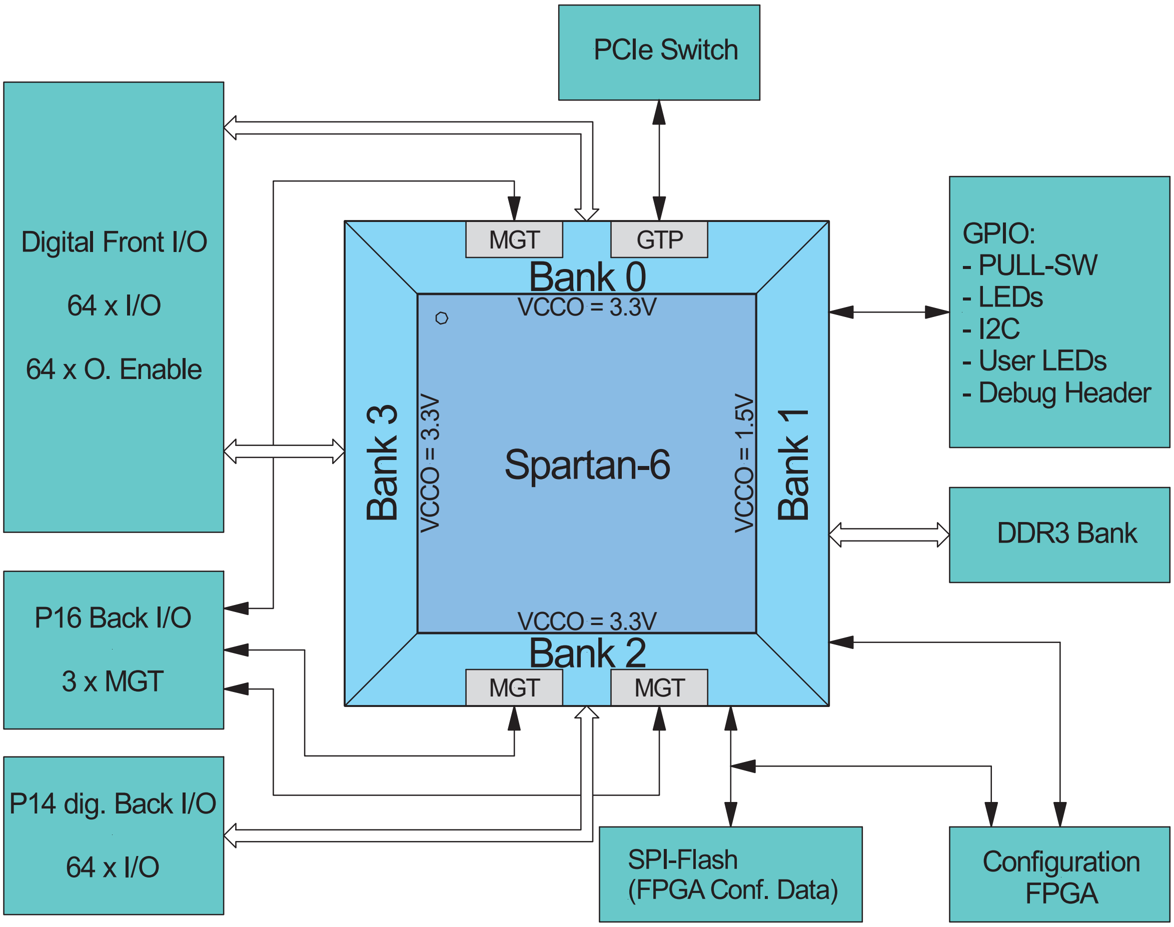


Figure 7-1 : FPGA Block Diagram

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 : TXMC633 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	V_{CCO}	V_{REF}	Signals	Remarks
Bank 0	3.3V	none	dig. Front I/O Interface	
Bank 1	1.5V	0.75V	DDR3 Bank Local Bus Interface pull-up/down config.	+GPIO / LED /Debug
Bank 2	3.3V	none	dig. Back I/O Interface	+Configuration
Bank 3	3.3V	none	dig. Front I/O Interface	
GTP Bank	**Description**			**Remarks**
MGT0	PCIe Endpoint Block to PCIe-Switch			
MGT1	MGT connection to XMC connector P16			
MGT2	MGT connection to XMC connector P16			
MGT3	MGT connection to XMC connector P16			

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 TXMC633 provides one MGT as Spartan-6 PCI Express Endpoint Block and three MGT for high speed XMC P16 interface.

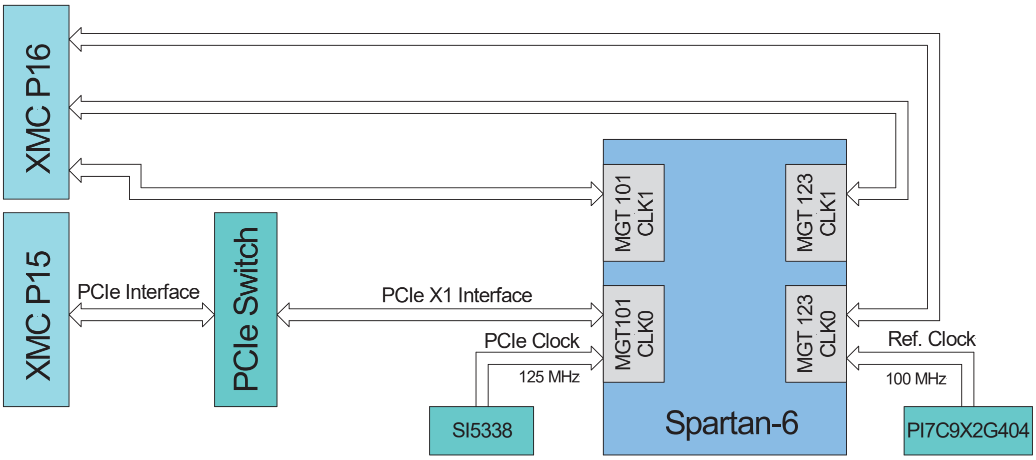


Figure 7-2 : GTP Block Diagram

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 TXMC633.

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 TXMC633 Spartan-6 FPGA always configures via x4 SPI Interface by “Master Serial / SPI” mode.

On delivery the SPI configuration Platform Flash contains the TEWS example application for the TXMC633 Spartan6 device.

7.4.1 Master Serial SPI Flash Configuration

It is important for User FPGA Configuration via ISP Master Mode that the ISP Mode Enable (ISP_EN) is set to disable the ISP Mode. This is the default value after the Power Up.
See also Register Description of TXMC633 Configuration Device.

7.4.2 Manually User FPGA SPI Flash Reconfiguration

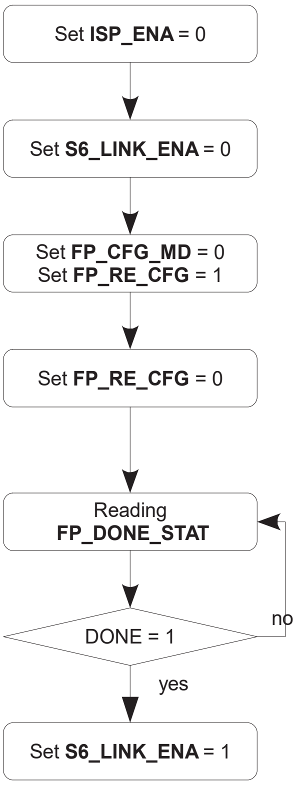
A manually User FPGA Reconfiguration could be release with User FPGA Reconfigure Command in the Global Configuration Register.

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 release 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 set and the Re-Configuration is prepared.
- Release a 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.

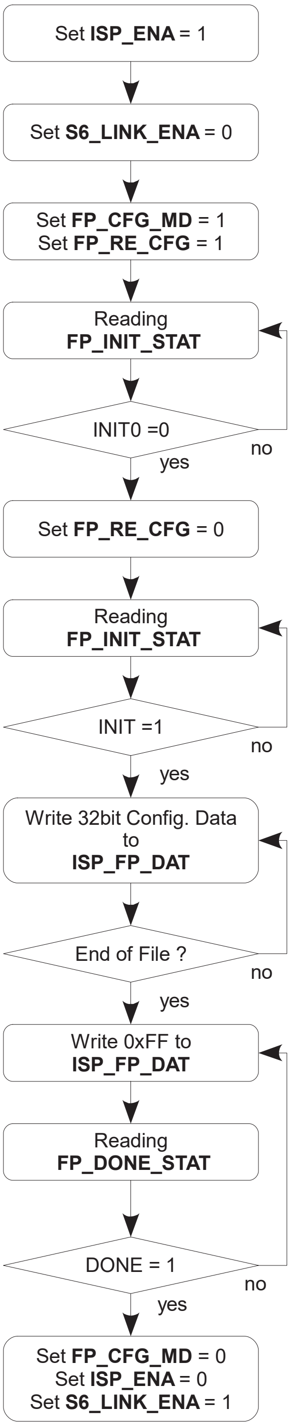
It must be considered in any case, that the Re-configuration of the User FPGA also Re-configure the PCIe Endpoint of the User FPGA. This has the consequence that the PCI Header of User FPGA PCIe Endpoint is no longer exists. For this purpose it is necessary to disable the link between the PCIe switch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

Addition, after Re-Configuration the User FPGA PCIe Endpoint the 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 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.
- Release a 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 necessary for 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.

An incomplete configuration could be aborted with the ISP_SPI_RST_CMD Command.

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 TXMC633 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   TXMC633
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   TXMC633
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 TXMC633 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.

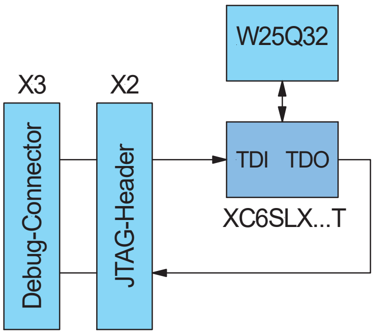


Figure 7-3 : User JTAG-Chain

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

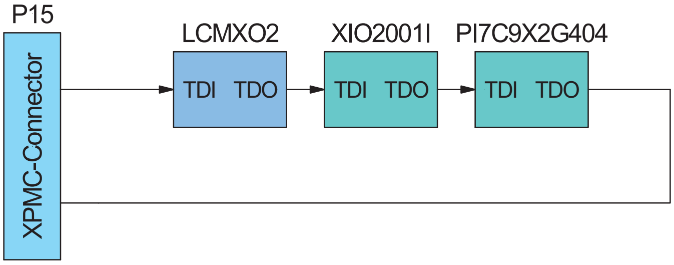


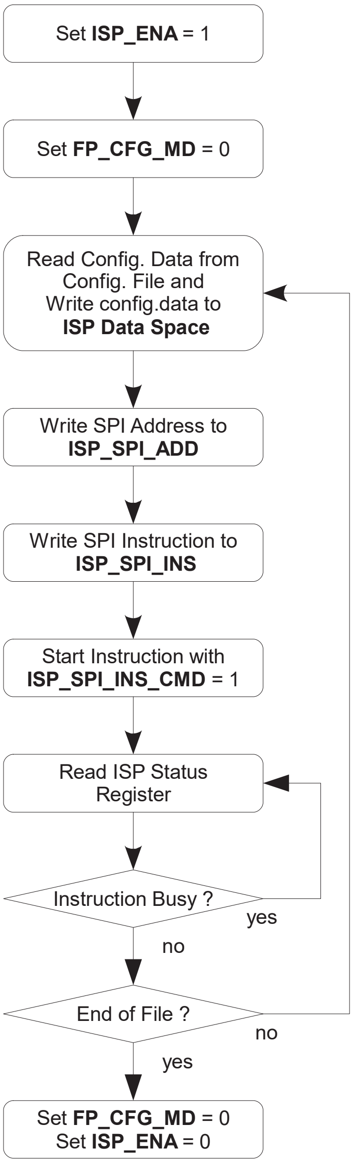
Figure 7-4 : TEWS Factory JTAG-Chain

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 released.



Set ISP_ENA = 1
Set FP_CFG_MD = 0
Read Config. Data from Config. File and Write config.data to ISP Data Space
Write SPI Address to ISP_SPI_ADD
Write SPI Instruction to ISP_SPI_INS
Start Instruction with ISP_SPI_INS_CMD = 1
Read ISP Status Register
Instruction Busy ?
yes
no
End of File ?
no
yes
Set FP_CFG_MD = 0
Set ISP_ENA = 0

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.

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

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

As a source for the User FPGA SPI Configuration Flash data should 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.

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

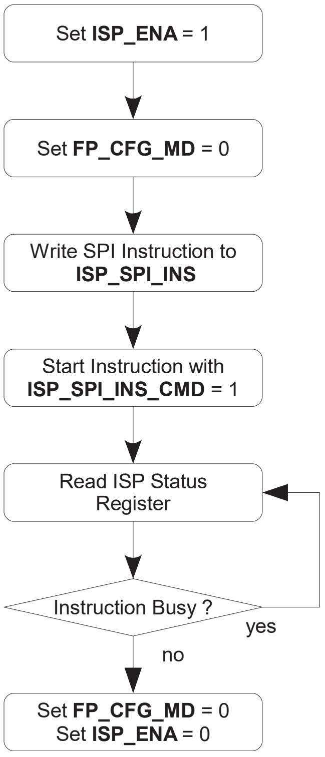
TXMC633-xx User FPGA SPI Configuration device:

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.

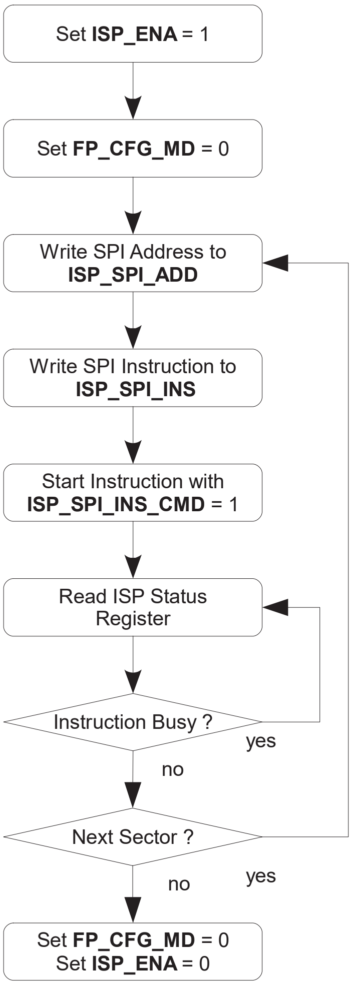
- Enable then ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash.
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for erasing process end.
- After completion 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.

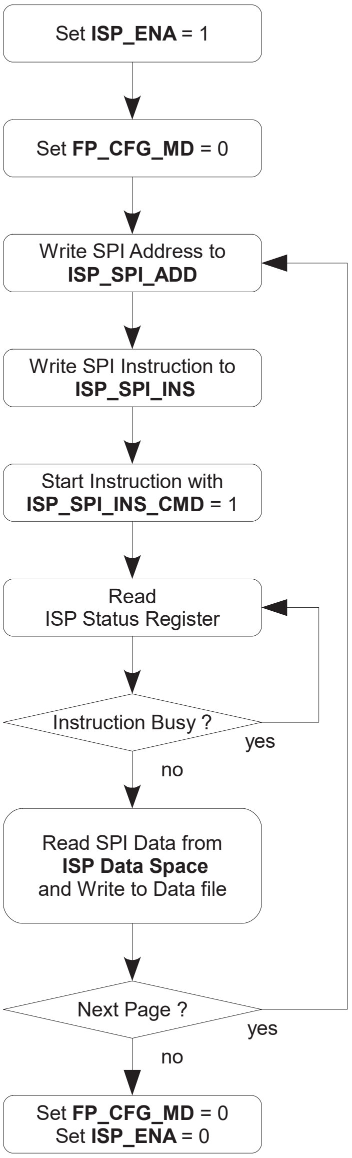
- Enable then ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash.
- Write Sector Address to the ISP Configuration Register
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait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


Set ISP_ENA = 1
Set FP_CFG_MD = 0
Write SPI Address to ISP_SPI_ADD
Write SPI Instruction to ISP_SPI_INS
Start Instruction with ISP_SPI_INS_CMD = 1
Read ISP Status Register
Instruction Busy ?
yes
no
Read SPI Data from ISP Data Space and Write to Data file
Next Page ?
yes
no
Set FP_CFG_MD = 0
Set ISP_ENA = 0

7.4.9 Generate Spartan6 Configuration Data

To use the maximum configuration speed, the TXMC633 must be configured to use the 32 MHz external master clock as CCLK.

To use this configuration feature, the following configuration option must be set:

'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

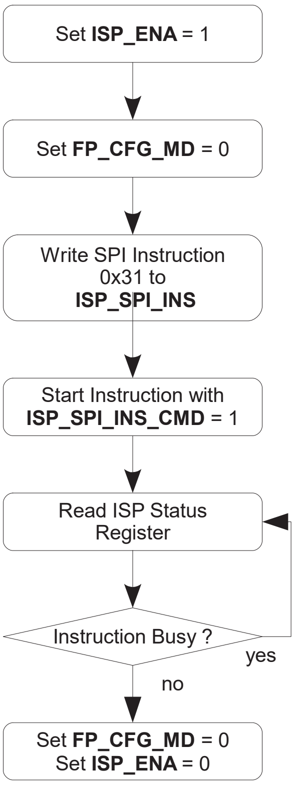
Without this option, the configuration time for the Spartan6 FPGA exceed the maximum PCIe bus setup time.

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. Therefor the SPI Flash on the TXMC633 provides a non-volatile register. This bit is always programmed 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 active the Spartan6 FPGA could not be configured from the SPI Flash.

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

- Enable then ISP Mode in the ISP Mode Enable Register.
- Assure that 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 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 TXMC633 FPGA configuration.

7.5 Clocking

7.5.1 FPGA Clock Sources

As a central clock generator of TXMC633 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.

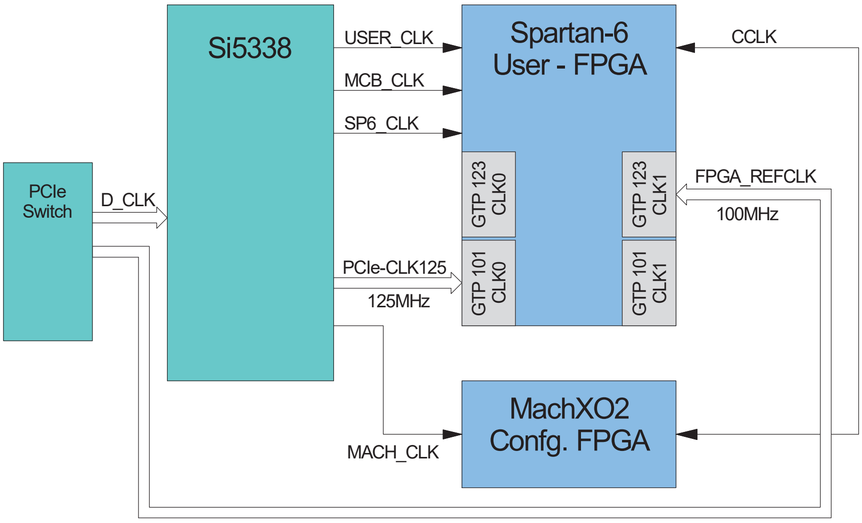


Figure 7-5 : FPGA Clock Sources

The following table lists the available clock sources on the TXMC633:

FPGA Clock-Pin Name	FPGA Pin Number	Source	Description
MGTREFCLK0_101	A10 / B10	SI5338 low-jitter clock generator	125 MHz 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 62.5 MHz
IO_L43P_GCLK22_3	M5	SI5338 low-jitter clock generator	USER CLK 83.3325 MHz
IO_L30N_GCLK0_2	AB13	SI5338 low-jitter clock generator	32.00 MHz Clock Input After configuration this clock could be used by FPGA design.
IO_L1P_CCLK_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 be to disable the 'spread-spectrum' feature in the host system.

7.6 Front I/O Interface

Each of the 64 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>#	F10	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_IO<48>#	L6	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<49>#	M7	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<50>#	H11	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<51>#	F15	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<52>#	H12	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<53>#	E16	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<54>#	G16	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<55>#	D19	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<56>#	K8	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<57>#	H13	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<58>#	G15	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<59>#	P7	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<60>#	F16	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<61>#	A19	IN/OUT	LVCMOS33	0	8	SLOW
FPGA_IO<62>#	N4	IN/OUT	LVCMOS33	3	8	SLOW
FPGA_IO<63>#	P3	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
FPGA_OE<48>#	K6	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<49>#	H8	OUTPUT	LVCMOS33	3	8	SLOW

FPGA_OE<50>#	M8	OUTPUT	LVCMOS33	3	8	SLOW
FPGA_OE<51>#	F14	OUTPUT	LVCMOS33	3	8	SLOW
FPGA_OE<52>#	G11	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<53>#	H14	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<54>#	B18	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<55>#	C19	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<56>#	G9	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<57>#	H10	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<58>#	A17	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<59>#	A18	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<60>#	C17	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<61>#	C18	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<62>#	A20	OUTPUT	LVCMOS33	0	8	SLOW
FPGA_OE<63>#	B20	OUTPUT	LVCMOS33	0	8	SLOW

Table 7-6 : Digital Front I/O Interface

7.7 Back I/O Interface

P14 Back I/O Pins of the TXMC633 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 TXMC633 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 TXMC633 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 TXMC633 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 TXMC633 provides a Winbond W25Q32 32-Mbit serial Flash memory, this Flash is used as FPGA configuration source (default configuration source). The TXMC633 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 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

7.9 Serial Number Allocation

The TXMC633 Module Serial Number is stored on-board on 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 needed to read the serial number via an I2C interface 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 A negative edge clock data out.
FPGA_SDA	1	1.5V	P18	Serial Data

Table 7-10 : 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 TXMC633 module	r	-

Table 7-11 : TXMC633 Serial Number

Example: 0x008F_DD0F => SNo.: 9428239

7.9.1 Device Addressing and Operation

The TXMC633 Configuration FPGA uses a standard 7 bit Slave Address. The eight bit of the slave address is the Read/write operation select bit.

1	0	1	0	1	0	1	R/$\overline{W}$
MSB							LSB

Figure 7-6 : Configuration FPGA Slave Address

TXMC633 Configuration FPFA I2C Slave typically Start and Stop condition

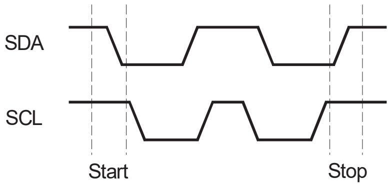


Figure 7-7 : Configuration FPGA Start and Stop condition

TXMC633 Configuration FPFA I2C Slave Output Acknowledge

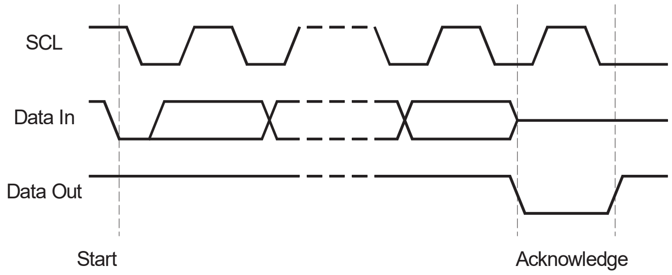


Figure 7-8 : Configuration FPGA Output Acknowledge

7.9.2 Read Operation

The TXMC633 Configuration FPGA provides only one 32 bit register which could be read via I2C Interface from User FPGA.

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.

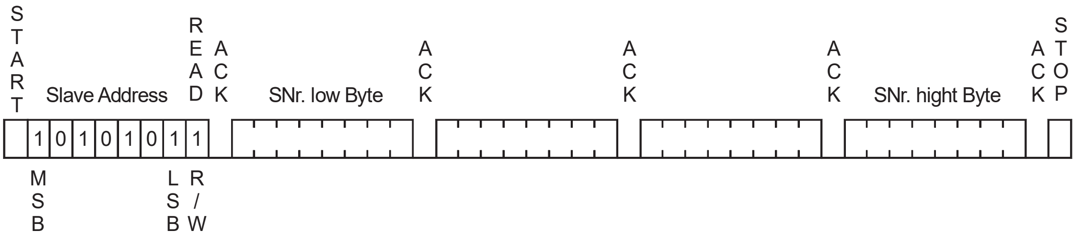


Figure 7-9 : Configuration FPGA Slave Access

7.9.3 Write Operation

A I2C write operation is not implemented.

7.10 I/O Pull Configuration

Each TTL I/O Line has a 4k7 Pull-Resistor. The 64 I/O Lines are divided into four 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 normal behaviour is that the User FPGA code controls the I/O Pull Configuration depending on User FPGA I/O Function.

The User FPGA (Spartan6) IO_PULL Interface configuration signals are connected via Configuration FPGA to four analog multiplexer. With these multiplexers the desired voltage can be adjusted directly from the User FPGA. Altogether there are eight controller lines which switch the four analog multiplexer for the I/O Pull Voltage. The user must therefore always ensure that valid signals are driven. Valid signals are driving zero for low level and for high level the line must be set to High-Z.

CNT **Lines**	**I/O Lines**	**Description**	**Spartan6 Pins**
IO_PULL[7:6]	IO_48 : IO_63	0bZZ : pull-down 0bZ0 : pull-up to 3.3V 0b0Z : pull-up to 5V 0b00 : No pull-up or pull-down	R16, R15
IO_PULL[5:4]	IO_31 : IO_47		M18, M17
IO_PULL[3:2]	IO_16 : IO_30		P16, N16
IO_PULL[1:0]	IO_00 : IO_15		T20, U19

Table 7-12 : I/O Pull Configuration

An additional option of setting the I/O Pull Configuration offers the Configuration FPGA. 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 wanted pull voltage.

7.11 User GPIO

The TXMC633 has some general purpose I/O and debug signals connected to User FPGA Bank 1. The required signaling standard is LVCMOS15, due to Memory Controller Block usage.

Two pins of the FPGA are routed to the Debug Connector for use as debug interface (UART). This is not a real RS-232 interface. A RS-232 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.

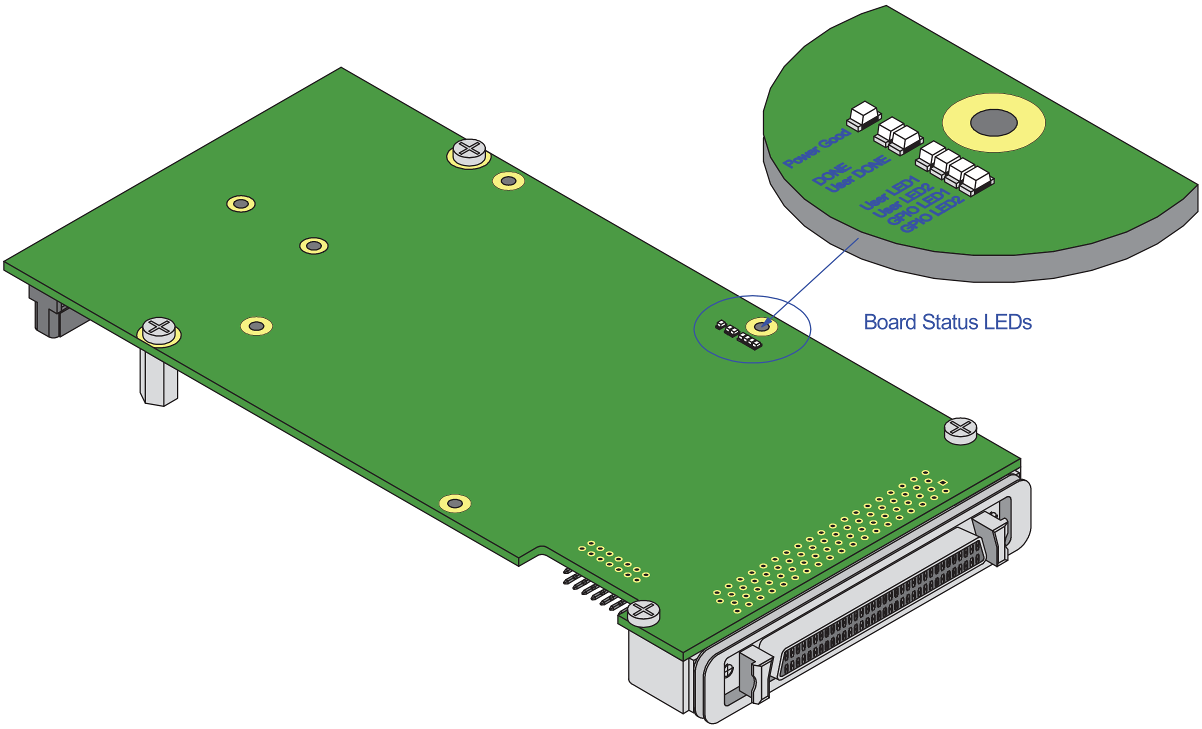
Also two free user programmable LEDs are connected to the User FPGA Bank 1.

Signal	Bank	V_{CCO}	Pin	Description
USER_LED0	1	1.5V	M16	2 x green on-board LEDs
USER_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 7-13 : FPGA General Purpose I/O

7.12 On-Board Indicators

The TXMC633 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.



LED	Color	Description
Power Good	Green	Power Good Signal for all on-board power supplies.
DONE	Green	Configuration FPGA DONE-Pin LED (MachXO2) Indicates successful FPGA configuration
User DONE	Green	User FPGA DONE-Pin LED (Spartan6) Indicates successful FPGA configuration
USER LED1 USER LED2	Green Green	Design dependent, can be controlled by the User FPGA. Refer to chapter “User-GPIO”
BCC_LED1 BCC_LED2	Green Green	Configuration FPGA depends.

Table 7-14 : Board-Status and User LEDs

7.13 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 as forced air cooling apply. Forced air cooling is recommended during operation.

The TXMC633 has a heat sink mounted on the Spartan-6 FPGA.

8 Design Help

8.1 Example Design

User applications for the TXMC633 can be developed using the TXMC633 FPGA Example Application design.

TEWS offers this FPGA Example design which consists of well documented 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 TXMC633. 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 TXMC633 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 TXMC633 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 64 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 analogue 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.

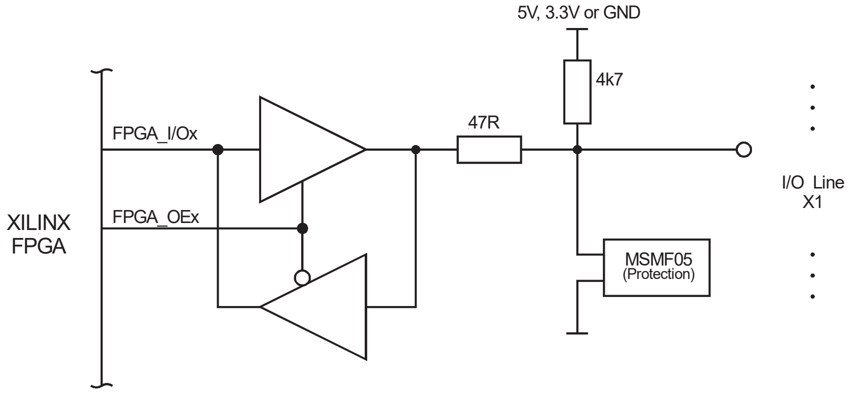


Figure 9-1 : TTL I/O Interface

Please note that the length (and consequently the capacitance) of a flat cable, connected to the TXMC633 module, should be kept as short as possible to prevent large cross talk.

To reduce the cross talk on the TXMC633, not all 64 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. Based on placement groups are needed for the pull voltage.

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.

9.1.1.1 Output Level & Output Current

Because of the 47 ohm series resistor, there is a reduced high-level voltage at the I/O pin when the output buffer sources a noticeable current to the external load while driving a high-level. To maintain a proper TTL high level, the recommended maximum I/O source current is 15mA.

Because of the 47 ohm series resistor, there is an increased low-level voltage at the I/O pin when the output buffer sinks a noticeable current from the external load while driving a low-level. To maintain a proper TTL low level, the recommended maximum I/O sink current is 6mA.

For achieving a 5V CMOS high-level voltage (V_{OH} >= 3.5V), the pull resistor reference may be set to 5V while the output transceiver is either actively driven low (low-level) or set to High-Z (5V high-level). However, this scenario is only applicable if the external load is high impedance. If there would be a low impedance path to ground on the I/O load, there may be a voltage divider with the on-board pull resistor, significantly reducing the high-level voltage at the I/O pin. To maintain a proper 5V CMOS high level, the I/O load (leakage) current should not exceed 250uA.

9.1.2 Differential I/O Interface

Each of the 32 (TXMC633-x1/x3) or 16 (TXMC633-x2/x4) differential I/O line pairs is realized with an I/O and output enable pin at the XILINX FPGA, connected to a differential I/O Buffer.

For TXMC633-x1/x2 a MAX3078E an ESD-protected RS485/RS422 transceiver, and a 120Ω termination resistor is provided.

For TXMC633-x3/x4 a SN65MLVD206 a Multipoint LVDS Line Driver and Receiver, and a 100Ω termination resistor is provided.

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

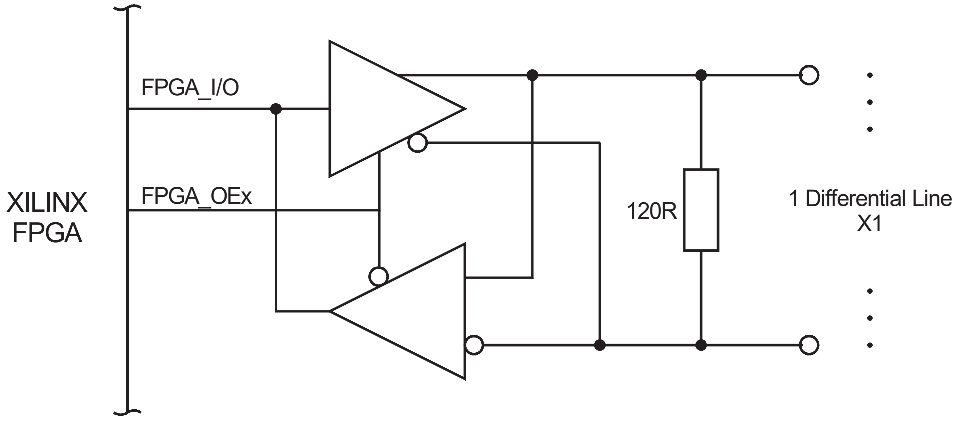


Figure 9-2 : Differential I/O Interface

Please consider that each TXMC633 M-LVDS line has its own termination. If more than four lines are connected together some termination resistors must be removed.

The actual data transmission rate depends on different factors like connection, cable length, FPGA design etc.

9.1.3 Back I/O Interface

P14 Back I/O Pins of the TXMC633 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 TXMC633 back I/O interface.

9.2 FPGA Debug Connector

The Debug Connector (X3) of the TXMC633 can be used to connect a debug adapter, if necessary. The debug adapter must be connected to the TXMC633 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.

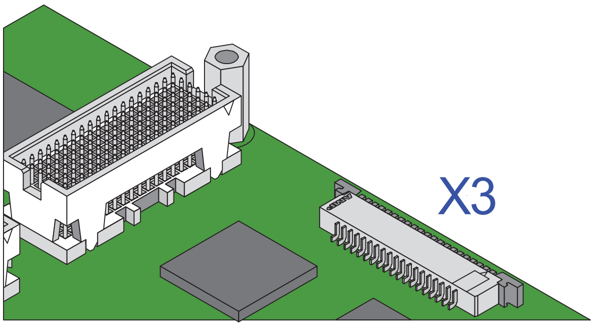
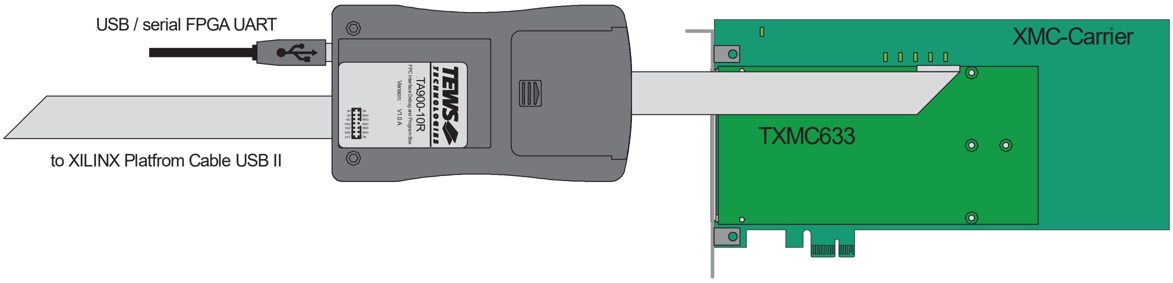


Figure 9-3 : Debug Connector X3

9.2.1 Connecting TA900 to TXMC633 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.

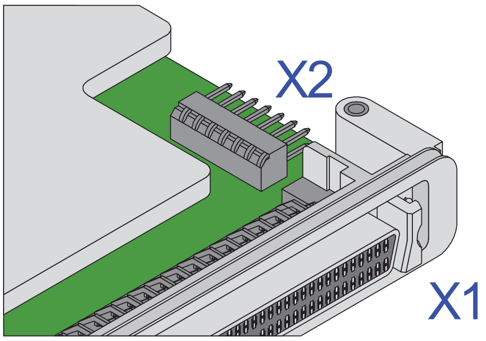
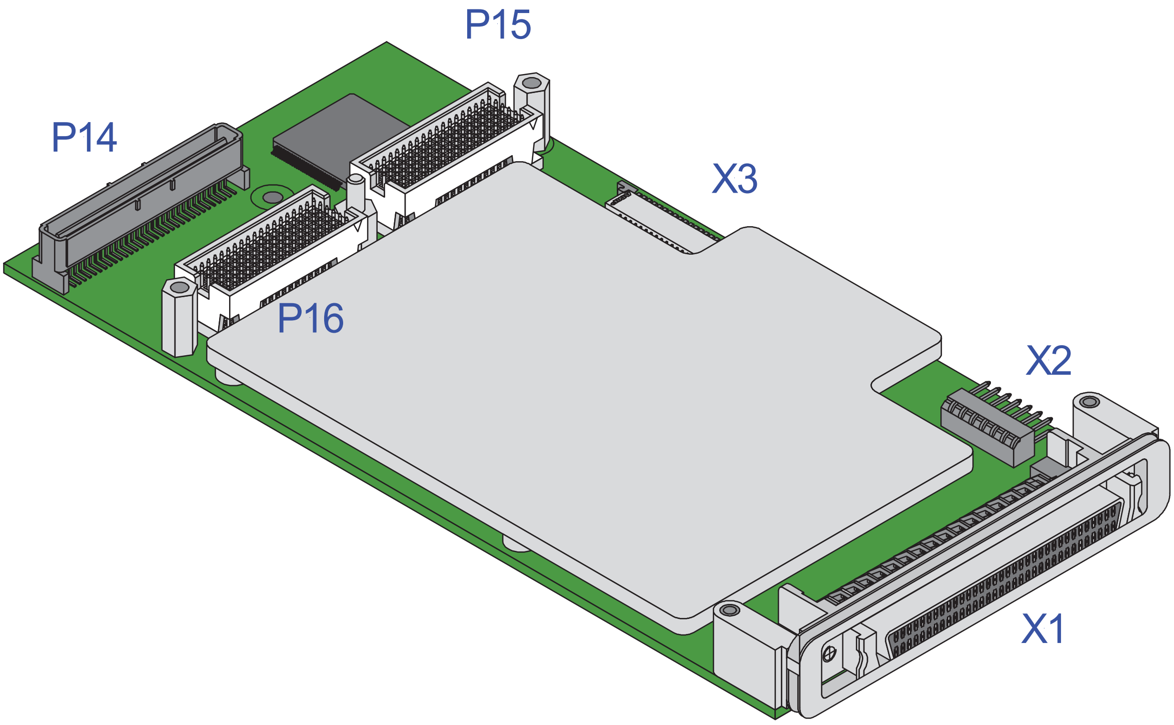


Figure 9-4 : FPGA JTAG Connector X2

10 Pin Assignment – I/O Connector

10.1 Overview



10.2 X1 Front Panel I/O Connector

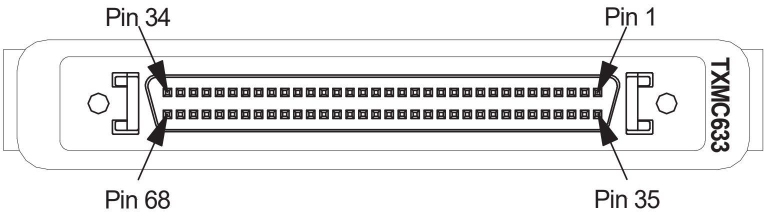


Figure 10-1 : Front Panel I/O Connector Numbering

10.2.1 Connector Type

Pin-Count	68
Connector Type	HD68 SCSI-3 type female connector
Source & Order Info	AMP 787082-7 or compatible

10.2.2 Pin Assignment

Pin	-x0	-x1 / -x3	-x2 / -x4
1	IO_0	IO_0A/-	IO_0A/-
2	IO_2	IO_1A/-	IO_1A/-
3	IO_4	IO_2A/-	IO_2A/-
4	IO_6	IO_3A/-	IO_3A/-
5	IO_8	IO_4A/-	IO_4A/-
6	IO_10	IO_5A/-	IO_5A/-
7	IO_12	IO_6A/-	IO_6A/-
8	IO_14	IO_7A/-	IO_7A/-
9	GND	GND	GND
10	IO_16	IO_8A/-	IO_8A/-
11	IO_18	IO_9A/-	IO_9A/-
12	IO_20	IO_10A/-	IO_10A/-
13	IO_22	IO_11A/-	IO_11A/-
14	IO_24	IO_12A/-	IO_12A/-
15	IO_26	IO_13A/-	IO_13A/-
16	IO_28	IO_14A/-	IO_14A/-
17	IO_30	IO_15A/-	IO_15A/-
18	IO_32	IO_16A/-	IO_32
19	IO_34	IO_17A/-	IO_34
20	IO_36	IO_18A/-	IO_36
21	IO_38	IO_19A/-	IO_38
22	IO_40	IO_20A/-	IO_40
23	IO_42	IO_21A/-	IO_42
24	IO_44	IO_22A/-	IO_44
25	IO_46	IO_23A/-	IO_46
26	GND	GND	GND
27	IO_48	IO_24A/-	IO_48
28	IO_50	IO_25A/-	IO_50
29	IO_52	IO_26A/-	IO_52
30	IO_54	IO_27A/-	IO_54
31	IO_56	IO_28A/-	IO_56
32	IO_58	IO_29A/-	IO_58
33	IO_60	IO_30A/-	IO_60
34	IO_62	IO_31A/-	IO_62
35	IO_1	IO_0B/+	IO_0B/+
36	IO_3	IO_1B/+	IO_1B/+
37	IO_5	IO_2B/+	IO_2B/+
38	IO_7	IO_3B/+	IO_3B/+
39	IO_9	IO_4B/+	IO_4B/+
40	IO_11	IO5B/+	IO5B/+
41	IO_13	IO6B/+	IO6B/+
42	IO_15	IO_7B/+	IO_7B/+
43	GND	GND	GND
44	IO_17	IO_8B/+	IO_8B/+
45	IO_19	IO_9B/+	IO_9B/+
46	IO_21	IO_10B/+	IO_10B/+
47	IO_23	IO_11B/+	IO_11B/+
48	IO_25	IO_12B/+	IO_12B/+
49	IO_27	IO_13B/+	IO_13B/+
50	IO_29	IO_14B/+	IO_14B/+
51	IO_31	IO_15B/+	IO_15B/+
52	IO_33	IO_16B/+	IO_33
53	IO_35	IO_17B/+	IO_35
54	IO_37	IO_18B/+	IO_37
55	IO_39	IO_19B/+	IO_39
56	IO_41	IO_20B/+	IO_41
57	IO_43	IO_21B/+	IO_43
58	IO_45	IO_22B/+	IO_45
59	IO_47	IO_23B/+	IO_47
60	GND	GND	GND
61	IO_49	IO_24B/+	IO_49
62	IO_51	IO_25B/+	IO_51
63	IO_53	IO_26B/+	IO_53
64	IO_55	IO_27B/+	IO_55
65	IO_57	IO_28B/+	IO_57
66	IO_59	IO_29B/+	IO_59
67	IO_61	IO_30B/+	IO_61
68	IO_63	IO_31B/+	IO_63

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

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-
29	BACK_IO14+		61	BACK_IO30+

Pin	differential I/O		Pin	differential I/O
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 TXMC633

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 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 TXMC633-xx

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 TXMC633
12	TRST#	not connected on the TXMC633
13	PGND	Used on TXMC633 for XILINX Header present detection
14	NC	HALT_INIT_WP signal. Optional. Not connected on the TXMC633

Table 10-2 : 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 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 TXMC633
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 TXMC633
15	3.3V	O	3.3V reference I/O voltage
16	FPGA_TxD	O	Used for Configuration FPGA on TXMC633
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 TXMC633.

Table 10-3 : Pin Assignment Debug Connector X3

11 Appendix A

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

```
## ############################################################################################## ##
##                                       TEWS TECHNOLOGIES                                        ##
## ############################################################################################## ##
##
##   Project Name    : TMXC633 UCF
##   File Name                : tmxc633.ucf
##   Target Device   : XC6SLXxxT-xFGG484
##   Design Tool     : Xilinx ISE Design Suit Embedded 14.7
##   Simulation Tool : -
##
##   Description     : The file lists all FPGA pins that are connected on the TXMC633
##
##   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 : (SE, 29.04.2014)
##                    Initial Version
##
##   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


## ############################################################################################# ##
## 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

net "MGTTX1_P"                        loc = "B8";           # Bank 101, XMC P16 MGT TX_P Lane 1
net "MGTTX1_N"                        loc = "A8";           # Bank 101, XMC P16 MGT TX_N Lane 1
net "MGTRX1_P"                        loc = "D9";           # Bank 101, XMC P16 MGT RX_P Lane 1
net "MGTRX1_N"                        loc = "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)

net "MGTTX2_P"                        loc = "B14";          # Bank 123, XMC P16 MGT TX_P Lane 2
net "MGTTX2_N"                        loc = "A14";          # Bank 123, XMC P16 MGT TX_N Lane 2
net "MGTRX2_P"                        loc = "D13";          # Bank 123, XMC P16 MGT RX_P Lane 2
net "MGTRX2_N"                        loc = "C13";          # Bank 123, XMC P16 MGT RX_N Lane 2
net "MGTTX3_P"                        loc = "B16";          # Bank 123, XMC P16 MGT RX_P Lane 3
net "MGTTX3_N"                        loc = "A16";          # Bank 123, XMC P16 MGT RX_N Lane 3
net "MGTRX3_P"                        loc = "D15";          # Bank 123, XMC P16 MGT RX_P Lane 3
net "MGTRX3_N"                        loc = "C15";          # Bank 123, XMC P16 MGT RX_N Lane 3

net "FPGA_REFCLK_P"                   loc = "A12";          # Bank 123, PCI Express Reference Clock
100 MHz (CLK_P)
                                                            #           via PI7C9X2G404 (PCI Express
Switch)
net "FPGA_REFCLK_N"                   loc = "B12";          # Bank 123, PCI Express Reference Clock
100 MHz (CLK_N)
                                                            #           via PI7C9X2G404 (PCI Express
Switch)


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

# Define I/O Standard
net "FPGA_OE[*]"               iostandard = LVCMOS33;       # Bank 0,3
net "FPGA_IO[*]"               iostandard = LVCMOS33;       # Bank 0,3
net "BACK_IO_P[*]"             iostandard = LVDS_33;        # Bank 2
net "BACK_IO_N[*]"             iostandard = LVDS_33;        # Bank 2


# 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_OE[48]"                     loc = "K6";           # Bank 3
net "FPGA_OE[49]"                     loc = "H8";           # Bank 3
net "FPGA_OE[50]"                     loc = "M8";           # Bank 3
net "FPGA_OE[51]"                     loc = "F14";          # Bank 0
net "FPGA_OE[52]"                     loc = "G11";          # Bank 0
net "FPGA_OE[53]"                     loc = "H14";          # Bank 0
net "FPGA_OE[54]"                     loc = "B18";          # Bank 0
net "FPGA_OE[55]"                     loc = "C19";          # Bank 0
net "FPGA_OE[56]"                     loc = "G9";           # Bank 0
net "FPGA_OE[57]"                     loc = "H10";          # Bank 0
net "FPGA_OE[58]"                     loc = "A17";          # Bank 0
net "FPGA_OE[59]"                     loc = "A18";          # Bank 0
net "FPGA_OE[60]"                     loc = "C17";          # Bank 0
net "FPGA_OE[61]"                     loc = "C18";          # Bank 0
net "FPGA_OE[62]"                     loc = "A20";          # Bank 0
net "FPGA_OE[63]"                     loc = "B20";          # 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 = "F10";          # Bank 0, 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
net "FPGA_IO[48]"                     loc = "L6";           # Bank 3, Front I/O X1
net "FPGA_IO[49]"                     loc = "M7";           # Bank 3, Front I/O X1
net "FPGA_IO[50]"                     loc = "H11";          # Bank 3, Front I/O X1
net "FPGA_IO[51]"                     loc = "F15";          # Bank 0, Front I/O X1
net "FPGA_IO[52]"                     loc = "H12";          # Bank 0, Front I/O X1
net "FPGA_IO[53]"                     loc = "E16";          # Bank 0, Front I/O X1
net "FPGA_IO[54]"                     loc = "G16";          # Bank 0, Front I/O X1
net "FPGA_IO[55]"                     loc = "D19";          # Bank 0, Front I/O X1
net "FPGA_IO[56]"                     loc = "K8";           # Bank 3, Front I/O X1
net "FPGA_IO[57]"                     loc = "H13";          # Bank 0, Front I/O X1
net "FPGA_IO[58]"                     loc = "G15";          # Bank 0, Front I/O X1
net "FPGA_IO[59]"                     loc = "P7";           # Bank 3, Front I/O X1
net "FPGA_IO[60]"                     loc = "F16";          # Bank 0, Front I/O X1
net "FPGA_IO[61]"                     loc = "A19";          # Bank 0, Front I/O X1
net "FPGA_IO[62]"                     loc = "N4";           # Bank 3, Front I/O X1
net "FPGA_IO[63]"                     loc = "P3";           # Bank 3, Front I/O X1

net "BACK_IO_P[0]"                    loc = "AA4";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[0]"                    loc = "AB4";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[1]"                    loc = "W6";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[1]"                    loc = "Y6";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[2]"                    loc = "T7";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[2]"                    loc = "U6";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[3]"                    loc = "Y7";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[3]"                    loc = "AB7";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[4]"                    loc = "V7";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[4]"                    loc = "W8";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[5]"                    loc = "AA8";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[5]"                    loc = "AB8";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[6]"                    loc = "T8";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[6]"                    loc = "U8";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[7]"                    loc = "R9";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[7]"                    loc = "R8";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[8]"                    loc = "Y9";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[8]"                    loc = "AB9";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[9]"                    loc = "U9";           # Bank 2, PMC Back I/O P14
net "BACK_IO_N[9]"                    loc = "V9";           # Bank 2, PMC Back I/O P14
net "BACK_IO_P[10]"                   loc = "AA10";         # Bank 2, PMC Back I/O P14
net "BACK_IO_N[10]"                   loc = "AB10";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[11]"                   loc = "W10";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[11]"                   loc = "Y10";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[12]"                   loc = "T10";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[12]"                   loc = "U10";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[13]"                   loc = "Y11";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[13]"                   loc = "AB11";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[14]"                   loc = "V11";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[14]"                   loc = "W11";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[15]"                   loc = "R11";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[15]"                   loc = "T11";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[16]"                   loc = "AA12";         # Bank 2, PMC Back I/O P14
net "BACK_IO_N[16]"                   loc = "AB12";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[17]"                   loc = "W12";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[17]"                   loc = "Y12";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[18]"                   loc = "T12";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[18]"                   loc = "U12";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[19]"                   loc = "V13";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[19]"                   loc = "W13";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[20]"                   loc = "AA14";         # Bank 2, PMC Back I/O P14
net "BACK_IO_N[20]"                   loc = "AB14";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[21]"                   loc = "W14";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[21]"                   loc = "Y14";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[22]"                   loc = "U14";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[22]"                   loc = "U13";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[23]"                   loc = "Y15";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[23]"                   loc = "AB15";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[24]"                   loc = "AA16";         # Bank 2, PMC Back I/O P14
net "BACK_IO_N[24]"                   loc = "AB16";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[25]"                   loc = "Y16";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[25]"                   loc = "W15";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[26]"                   loc = "U16";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[26]"                   loc = "V15";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[27]"                   loc = "T15";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[27]"                   loc = "U15";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[28]"                   loc = "Y17";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[28]"                   loc = "AB17";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[29]"                   loc = "AA18";         # Bank 2, PMC Back I/O P14
net "BACK_IO_N[29]"                   loc = "AB18";         # Bank 2, PMC Back I/O P14
net "BACK_IO_P[30]"                   loc = "W17";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[30]"                   loc = "Y18";          # Bank 2, PMC Back I/O P14
net "BACK_IO_P[31]"                   loc = "V17";          # Bank 2, PMC Back I/O P14
net "BACK_IO_N[31]"                   loc = "W18";          # Bank 2, PMC Back I/O P14


## ############################################################################################# ##
## 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 Constratints
config mcb_performance                    = standard;       # General MCB constraints


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

# I/O Standards
net "MCB_CLK"                  iostandard = LVCMOS33;       # Bank 2
net "SP6_CLK"                  iostandard = LVCMOS33;       # Bank 2
net "USER_CLK"                 iostandard = LVCMOS33;       # Bank 3


# Location Constraints
net "SP6_CLK"                         loc = "AB13";         # Bank 2
net "MCB_CLK"                         loc = "Y13";          # Bank 2
net "USER_CLK"                        loc = "M5";           # Bank 3
```

```
# Additional Constraints
net "SP6_CLK"                     tnm_net = "SP6_CLK";
timespec "TS_SP6_CLK"                     = period "SP6_CLK" 32 MHz high 50 %;
net "MCB_CLK"                     tnm_net = "MCB_CLK";
timespec "TS_MCB_CLK"                     = period "MCB_CLK" 62.5 MHz high 50 %;
net "USER_CLK"                    tnm_net = "USER_CLK";
timespec "TS_USER_CLK"                    = period "USER_CLK" 83.3325 MHz high 50 %;


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

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


# Location Constraints
net "LL_BUS[0]"                       loc = "B22";          # Bank 1
net "LL_BUS[1]"                       loc = "J16";          # Bank 1
net "LL_BUS[2]"                       loc = "J17";          # Bank 1
net "LL_BUS[3]"                       loc = "C20";          # Bank 1
net "LL_BUS[4]"                       loc = "C22";          # Bank 1
net "LL_BUS[5]"                       loc = "L15";          # Bank 1
net "LL_BUS[6]"                       loc = "K16";          # Bank 1
net "LL_BUS[7]"                       loc = "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 "PULL_IN[6]"                      loc = "R15";          # Bank 1, Group [48 : 63]
net "PULL_IN[7]"                      loc = "R16";          # Bank 1, Group [48 : 63]

net "DWNRST2_n"                       loc = "L3";           # Bank 3, PCI Express Reset


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

# I/O Standards
net "USER_LED[?]"              iostandard = LVCMOS15;       # Bank 1
net "FPGA_SDA"                 iostandard = LVCMOS15;       # Bank 1
net "FPGA_SCL"                 iostandard = LVCMOS15;       # Bank 1
net "USER_RXD"                 iostandard = LVCMOS15;       # Bank 1
net "USER_TXD"                 iostandard = LVCMOS15;       # Bank 1
net "USER_BUT"                 iostandard = LVCMOS15;       # Bank 1, Input Only


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

net "FPGA_SDA"                        loc = "P18";          # Bank 1, Debug Connector X3
net "FPGA_SCL"                        loc = "R17";          # Bank 1, Debug Connector X3

net "USER_RXD"                        loc = "T17";          # Bank 1, Debug Connector X3
net "USER_TXD"                        loc = "T19";          # Bank 1, Debug Connector X3
net "USER_BUT"                        loc = "T18";          # Bank 1, Debug Connector X3
```