The Embedded I/O Company

TEWS TECHNOLOGIES

TXMC638

Reconfigurable FPGA with 24 x 16 Bit Analog Input

Version 1.0

User Manual

Issue 1.0.2

October 2017

powerBridge Computer

Ehlbeek 15a
30938 Burgwedel
fon 05139-9980-0
fax 05139-9980-49

www.powerbridge.de
info@powerbridge.de

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

TXMC638-10R

24 x Analog In and 64 direct FPGA Back I/O Lines,

XC7K160T-2 FBG676 Kintex-7 FPGA, 1GB DDR3

TXMC638-11R

24 x Analog In and 64 direct FPGA Back I/O Lines,

XC7K325T-2 FBG676 Kintex-7 FPGA, 1GB DDR3

TXMC638-12R

24 x Analog In and 64 direct FPGA Back I/O Lines,

XC7K410T-2 FBG676 Kintex-7 FPGA, 1GB DDR3

Style Conventions

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

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

Access terms are described as:

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

Issue	Description	Date
1.0.0	Initial issue	October 2016
1.0.1	Additions to the technical specifiction and correction of the MGT Connections Table. User FPGA Configuration Flow Charts enhanced.	November 2016
1.0.2	Insert the missing Flow Charts in Chapter 7.4.2 to 7.4.8.	October 2017

Table of Contents

List of Figures

List of Tables

1 Product Description

The TXMC638 is a standard single-width Switched Mezzanine Card (XMC) compatible module providing a user configurable Xilinx Kintex-7 FPGA with 24 ADC input channel.

The TXMC638 provides 24 ADC input channels based on the Linear Dual 16-Bit 5Msps Differential LTC2323-16 ADC. Each of the 24 channels has a resolution of 16bit and can work with up to 5Msps. The analog input circuit is designed to allow input voltages up to ±2.5 V on each input-pin (results in ±5 V differential voltage range)

For customer specific I/O extension or inter-board communication, the TXMC638 provides 64 FPGA I/Os on P14 and 4 FPGA Multi-Gigabit-Transceiver on P16. P14 I/O lines can be configured as 64 single ended LVCMOS25 or as 32 differential LVDS25 interface.

Additionally the TXMC638 provides three 100 Ohm terminated ac-coupled, differential inputs with wide Input voltage range.

The User FPGA is connected to a 1GB, 32 bit wide DDR3 SDRAM. The SDRAM-interface uses an internal Memory Controller of the Kintex-7.

The User FPGA is configured by a serial SPI flash. For full PCIe specification compliance, the XILINX Tandem Configuration Feature can be used for FPGA configuration. XILINX Tandem Methodologies “Tandem PROM” should be the favored Methodology. The SPI 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 by using the Xilinx Vivado Logic Analyzer.

User applications for the TXMC638 with Kintex-7 FPGA can be developed using the Xilinx design software Vivado Design Suite. A license for the Vivado Design Suite design tool is required.

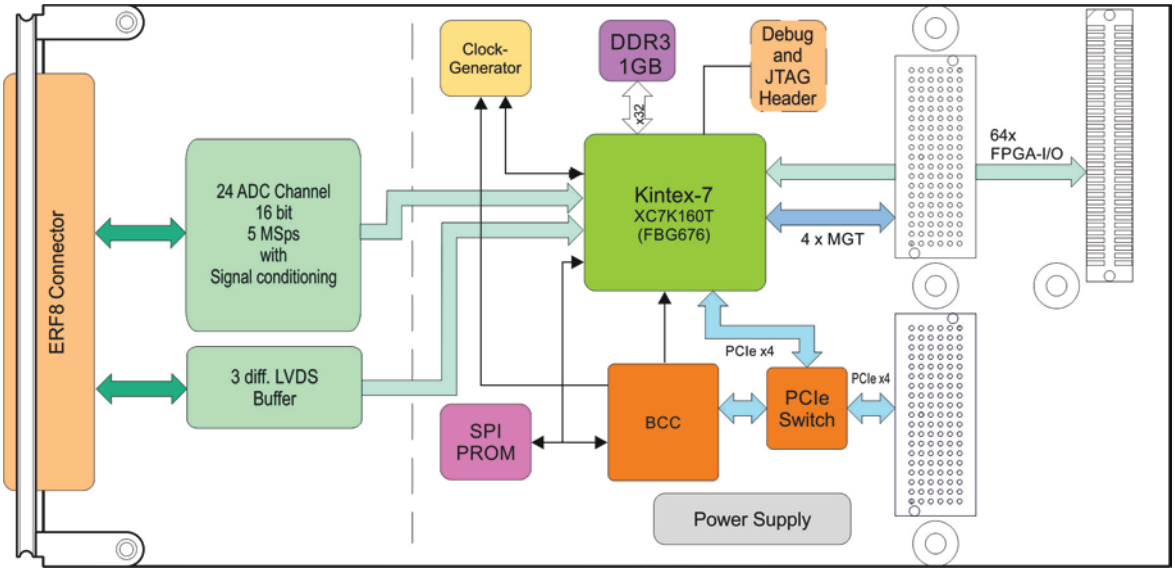


Figure 1-1 : Block Diagram

2 Technical Specification

XMC Interface		
Mechanical Interface	Switched Mezzanine Card (XMC) Interface confirming to ANSI/VITA 42.0-2008 (Auxiliary Standard) Standard single-width (149mm x 74mm)	
Electrical Interface	PCI Express x4 Link (Base Specification 1.1) compliant interface conforming to ANSI/VITA 42.3-2006 (XMC PCI Express Protocol Layer Standard)	
On-Board Devices		
PCI Express Switch	PI7C9X2G312GP (Pericom)	
PCI Express to PCI Bridge	XIO2001 (Texas Instruments)	
User configurable FPGA	TXMC638-10R: XC7K160T-2FBG676I (Xilinx) TXMC638-11R: XC7K325T-2FBG676I (Xilinx) TXMC638-12R: XC7K410T-2FBG676I (Xilinx)	
SPI-Flash	N25Q128A (Micron) 128 Mbit (contains TXMC638 FPGA Example) or compatible; +3.3V supply voltage	
DDR3 RAM	2 x MT41K256M16HA-125 (Micron) 256 Meg x 32 Bit	
Board Configuration FPGA	LCMXO2-7000HC (Lattice)	
ADC	LTC2323IUFD-16 (Linear Technologies)	
I/O Interface		
Number of analog Input	24 differential 16 bit Inputs	
Analog Input Voltage	diff. VIN_{MAX} (allowed voltage between input pins): Common Mode Voltage Range : Input Voltage limit for each pin relative to common ground	±5.0V ±7.5V ±10V
Number of digital Front I/O	Three differential Front I/O Inputs with wide Input voltage range. Differential voltage range: ±200mV up to ±3.6V Differential Termination = 100 Ohm	
Number of digital Back I/O	64 direct FPGA I/O lines to P14 Back I/O connector - Possible use as single ended or differential I/O - FPGA I/O Standard: LVCMOS25, LVTTL25 and LVDS_25	
I/O Connector	Front I/O Samtec – ERF8-049-01-L-D-RA-L PMC P14 I/O (64 pin Mezzanine Connector) XMC P16 I/O (114 pin Mezzanine Connector)	
Physical Data		
Power Requirements	Depends on FPGA design With TXMC638 Board Reference Design / without external load	

	typical @ +5V VPWR	typical @ +12V VPWR
TXMC638-xxR	3.0 A	1.25 A

Temperature Range	Operating Storage	-40°C to +85 °C -40°C to +85°C

MTBF	TXMC638-xxR: 269000 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	TXMC638-xxR: 133g

Table 2-1 : Technical Specification

3 Handling and Operation Instruction

3.1 ESD Protection

The TXMC638 is sensitive to static electricity. Packing, unpacking and all other handling of the TXMC638 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.

Please also note chapter “Thermal Management”.

3.3 Assembling Hints

When disassembling the TXMC638 from carrier board please keep the mechanical stress as low as possible.

4 PCI Device Topology

The TXMC638 consists of two FPGAs. Both FPGA are designed as a PCIe / PCI endpoint devices. One FPGA is the User FPGA (Kintex-7) which could be programmed with user defined FPGA code. The second FPGA takes control of on-board hardware functions of TXMC638 and also the configuration control of the User FPGA. This second FPGA is the BCC (Board Configuration Controller).

The BCC PCI endpoint is connected via a PCI-to-PCIe Bridge to the second x1 Downstream Port of the PCIe Switch (Pericom PI7C9X2G312GP). The User FPGA (Kintex-7 PCIe endpoint) is directly connected to the first x4 Downstream Port.

The x4 Upstream Port of the PCIe Switch is connected to the XMC P15 Connector, communicating with the host system.

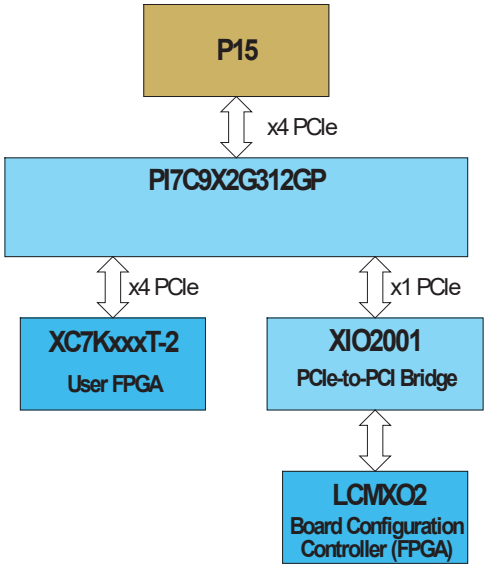


Figure 4-1 : PCIe/PCI Device Topology

Device	Vendor ID	Device ID	Class Code	Description (as shown by lspci)
PI7C9X2G312GP	0x12D8 (Pericom)	0x2312	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
XC7KxxxT-2	user defined			Device identification for the User programmable FPGA is defined by user. The data will be created with the Xilinx Vivado “7 Series Integrated Block for PCI Express” IP generator.
BCC LCMXO2	0x1498 (TEWS)	0x927E	0x068000	Bridge Device: TEWS Technologies GmbH Device 927E (TXMC638).

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

4.1 User FPGA (Kintex-7)

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

4.2 BCC (Board Configuration Controller) FPGA

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	927E 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	927E 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	-
0xCC	Reserved	-
0xD0	User FPGA Configuration Control/Status Register	32
0xD4	User FPGA Configuration Data Register (Slave SelectMAP)	32
0xD8	Reserved	-
0xDC	Reserved	-
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	Reserved	-
0xF8	TXMC638 Serial Number	32
0xFC	BCC Code Version	32

Table 4-4 : Local Configuration Register Space

4.2.2.2 In-System Programming Data Space

The In-System Programming (ISP) 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 ISP are located in the Local Configuration Register Space. The data register for direct FPGA ISP is also located in the Local Configuration Register Space.

5 Register Description

5.1 User FPGA (Kintex-7)

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

5.2 Board Configuration Controller (BCC - FPGA)

5.2.1 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 5-1 : Interrupt Enable Register

5.2.2 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 5-2 : Interrupt Status Register

5.2.3 User FPGA Configuration Control/Status Register - 0xD0

Bit	Symbol	Description	Access	Reset Value
31:5		Reserved		0
4	K7_LINK_ENA	1: Kintex-7 to PCIe-Switch LINK is enabled 0: Kintex-7 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 5-3 : User FPGA Configuration Control/Status Register

5.2.4 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 Slave Select Map 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 5-4 : User FPGA Configuration Data Register

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

5.2.5 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 the BCC interface between BCC, SPI-Flash and the User FPGA (Kintex-7). When set, the BCC is both SPI Flash Master and FPGA Configuration Interface Master. Must be set to 1 for direct Slave Select Map mode 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 Select Map configuration mode.	R/W	0

Table 5-5 : ISP Control Register

5.2.6 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 0xC7 – Chip Erase 0x03 – Read Data	w	0x00

Table 5-6 : ISP Configuration Register

5.2.7 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 5-7 : ISP Command Register

5.2.8 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 5-8 : ISP Status Register

5.2.9 TXMC638 Serial Number - 0xF8

Bit	Symbol	Description	Access	Reset Value
31:0	S_NUMBER	The value is the unique serial number of each TXMC638 module	R	-

Table 5-9 : TXMC638 Serial Number

Example: 0x0091_981A => SNo.: 9541658

The serial number can also be read via an I2C interface from User FPGA (KIntex-7).

5.2.10 BCC - FPGA Code Version - 0xFC

Bit	Symbol	Description	Access	Reset Value
31:0	CODE_VER	The value shows the BCC Firmware code version of the TXMC638 module.	R	-

Table 5-10: BCC - FPGA Code Version

Example:

0x0100_0A00 => bit 32 downto 24 : Major FPGA Code Version

0x0100_0A00 => bit 23 downto 16 : Minor FPGA Code Version

0x0100_0A00 => bit 15 downto 08 : FPGA Code Revision

0x0100_0A00 => bit 07 downto 00 : FPGA Code Build Number

6 Interrupts

6.1 Interrupt Sources

6.1.1 User FPGA (Kintex-7)

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

6.1.2 Board Configuration Controller (BCC - FPGA)

The BCC - 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 (Kintex-7)

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

6.2.2 Board Configuration Controller (BCC - FPGA)

Both Interrupts of the BCC 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

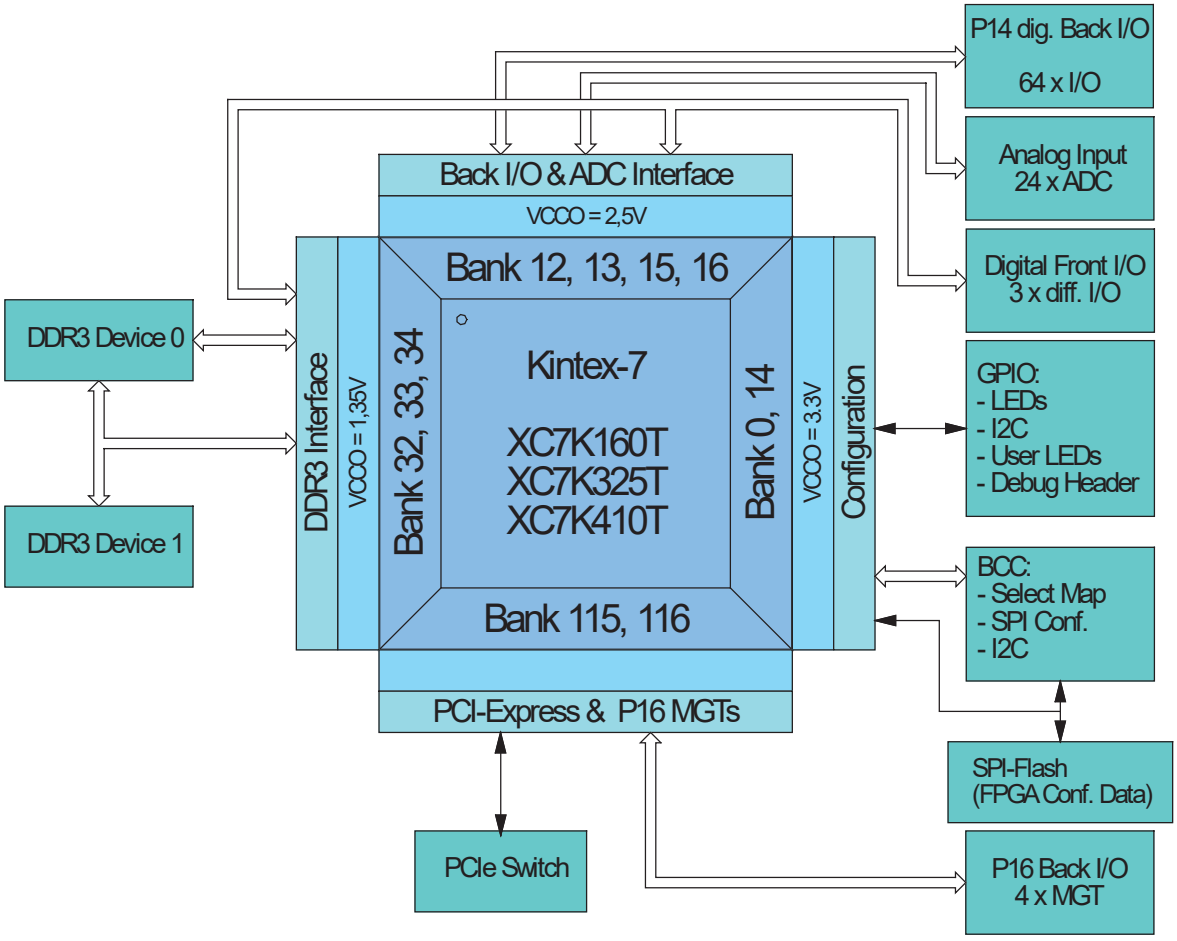


Figure 7-1 : FPGA Block Diagram

7.2 User FPGA Highlights

The FPGA is a Kintex-7 XC7K160T, XC7K325T or XC7K410T FPGA. Each Kintex-7 FPGA in a FBG676 package provides eight GTX four for high speed back I/O communication and four for the PCI Express interface (x4 Linkage).

KIntex-7	Logic Cells	Slices	DSP Slices	Block RAM (Kb)			CMTs	GTXs (MGT)	XADC Block
				18 Kb	36 Kb	max(Kb)			
XC7K160T	162,240	25.350	600	650	325	11,700	8	8	1
XC7K325T	326,080	50.950	840	890	445	16,020	10	8	1
XC7K410T	406,720	63.550	1540	1590	795	28,620	10	8	1

Table 7-1 : TXMC638 FPGA Feature Overview

PCI Express Highlights:

- Compliant to the PCI Express Base Specification 2.1 with Endpoint and Root Port capability.
- Supports Gen1 (2.5 Gb/s) and Gen2 (5 Gb/s)

XADC Highlights:

XADC (Analog-to-Digital Converter)

On-chip temperature (±4°C max error) and power supply (±1% max error) sensors

- Continuous JTAG access to ADC measurements
- Internal access to all internal sensors of the Kintex-7

The board supports JTAG, master serial mode configuration from SPI-Flash or Slave Select MAP configuration for the User FPGA (Kintex-7) via Board Configuration Controller (BCC).

The User FPGA is equipped with 6 I/O banks and 8 GTX (Gigabit Transceiver).

Bank	VCCO	VREF	Signals	Note
Bank 0	3.3V	none	SPI Configuration with or without Tandem configuration Slave Select Map Configuration GPIOs	
Bank 14	3.3V	none		
Bank 12	2.5V	none	ser. ADC Interface Back I/O diff. digital Front I/O (Ch.0 & 1)	
Bank 13	2.5V	none		
Bank 15	2.5V	none		
Bank 16	2.5V	none		
Bank 32	1.35V	0.625V	DDR3 Memory Interface 1GB diff. digital Front I/O (Ch. 2)	
Bank 33	1.35V	0.625V		
Bank 34	1.35V	0.625V		
Bank 115	Optional used for Back I/O link			
Bank 116	PCIe X4 Interface to PCIe Switch Device			

Table 7-2 : FPGA Bank Usage

7.3 User FPGA Gigabit Transceiver (MGT)

The TXMC638 provides four MGT as Kintex-7 PCI Express Endpoint Block and four MGT for high speed XMC P16 interface.

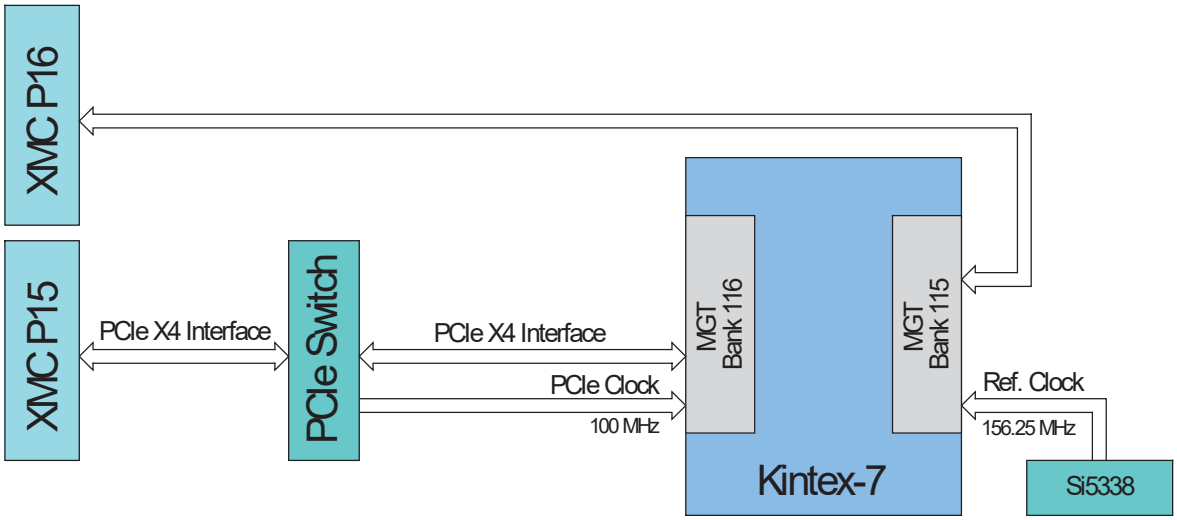


Figure 7-2 : GTP Block Diagram

GTP	Signal	FPGA Pins	Connected to
MGTXTXP0_115	MGTTX0	P2 / P1	connected to XMC P16
	MGTRX0	R4 / R3	
MGTXTXP1_115	MGTTX1	M2 / M1	
	MGTRX1	N4 / N3	
MGTXTXP2_115	MGTTX2	K2 / K1	
	MGTRX2	L4 / L3	
MGTXTXP3_115	MGTTX3	H2 / H1	
	MGTRX3	J4 / J3	
MGTXTXP0_116	PET07	G4 / G3	used for PCI Express Endpoint Block
	PER07	F2 / F1	
MGTXTXP1_116	PET06	E4 / E3	
	PER06	D2 / D1	
MGTXTXP2_116	PET05	C4 / C3	
	PER05	B2 / B1	
MGTXTXP3_116	PET04	B6 / B5	
	PER04	A4 / A3	

Table 7-3 : MGT Connections

The MGT clock MGTREFCLK0_116 (PCI Express Endpoint Block clock reference) of 100 MHz is generated by the PI7C9X2G312GP PCIe Switch. The MGTREFCLK0_115 is connected to a 156.25 MHz clock output of the Si5338 low jitter clock generator. MGTREFCLK1_115 and MGTREFCLK1_116 are not used on the TXMC638.

GTP	Signal	FPGA Pins	Connected to
MGTREFCLK0_115	CLK_MGT	H6 / H5	156.25 MHz Si5338 Clock Generator
MGTREFCLK1_115	not used	K6 / K5	not connected
MGTREFCLK0_116	REFCLKO2	D6 / D5	100 MHz PI7C9X2G312GP PCIe Switch
MGTREFCLK1_116	not used	F6 / F5	not connected

Table 7-4 : Multi Gigabit Transceiver Reference Clocks

7.4 User FPGA Configuration

The Kintex-7 could be configured by the following interfaces:

- Master Serial SPI Flash Configuration Interface
- JTAG Interface via FPGA JTAG Connector
- PCIe Interface via BCC FPGA Slave Select Map Interface Configuration

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

At Power-up, the TXMC638 User FPGA (Kintex-7) always configures via x4 SPI Interface by "Master Serial / SPI" mode.

On delivery the SPI Flash contains the TEWS example application for the TXMC638 User FPGA device.

7.4.1 Master Serial SPI Flash Configuration

It is important for User FPGA Configuration via SPI Master Mode that the ISP Mode Enable (ISP_EN) bit is clear to disable the ISP Mode. This is also the default value after Power Up.
See also Register Description of TXMC638 Configuration Device.

To comply with the PCI-Express specification it is necessary to perform the configuration as quick as possible. The PCIe specification dictates that a PCI device must be accessible after 100ms (120ms). To speed up the SPI Configuration the following points must be taken into account for SPI Bitstream generation.

- External Clock Master (53.2MHz) should be used.
- If external Clock Master is used, also the SPI Falling Edge Option must be used.
- SPI Configuration Bus Width should be set to X4.
- Xilinx Tandem Configuration Feature could be used for full PCI-Express specification compliance. Already during PCI-Express IP Core generation this configuration feature must be included. (For more information see: XAPP1179).
- If the Tandem Configuration feature is used, the Persist Option is mandatory.
- For smaller FPGA content, it is sometimes also possible to comply with the PCI-Express specification, when only Bitstream Compression is used.

To avoid damage on the BCC or User FPGA (Kintex-7) if Tandem configuration or the Persist Option is used, the User FPGA must be set into reconfigure Mode by using the "FP_RE_CFG" Bit of the User FPGA Configuration Control/Status Register before Programming or Clearing the SPI Flash.

7.4.2 Manually User FPGA SPI Flash Reconfiguration

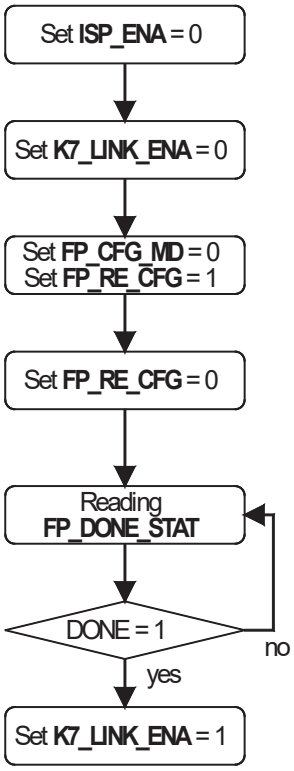
A manually User FPGA Reconfiguration could be performed with the User FPGA Reconfigure Command in the Global Configuration Register.

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

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

- Assure that ISP Mode Enable is disabled.
- By Re-configuring the Kintex-7 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 Kintex-7 is disabled.
- Set the User FPGA Configuration Mode (FP_CFG_MD) to Master Serial / SPI and prepare the FPGA Re-Configuration.
- Start the FPGA Re-configuration by setting the FP_RE_CFG bit of the User FPGA Configuration Control/Status Register to 0.
- Assure that the FPGA DONE Pin status shows a successful FPGA Configuration.

  0: FPGA DONE Pin Level is Low (FPGA is not configured)
  1: FPGA DONE Pin Level is High (FPGA is configured)
- The link between the PCIe Switch and the Kintex-7 must be enabled.



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

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

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

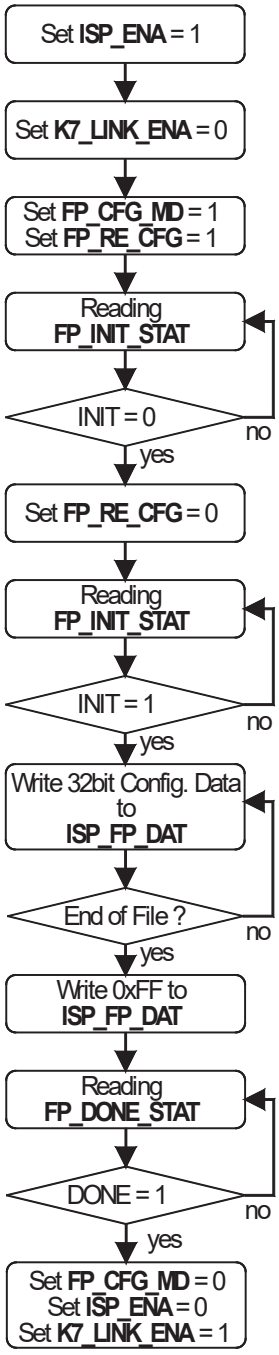
7.4.3 Slave Select Map Configuration

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

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

- First the In System Program (ISP) Mode must be enabled.
- By Re-configuring the Kintex-7 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 Kintex-7 is disabled.
- Check response of the Kintex-7 by reading the FPGA INIT_B pin value. If the Level is low the Kintex-7 FPGA is in Reset Mode, and then configuration process could be continued.
- Start the FPGA Re-configuration by setting the FP_RE_CFG bit of the User FPGA Configuration Control/Status Register to 0.
- Check response of the Kintex-7 by reading the FPGA INIT_B pin value. While the FPGA INIT_B pin Level is low the Kintex-7 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 2860903 PCI write accesses are required to configure a Kintex-7 325T.
- 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 Kintex-7 must be enabled.



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

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

The .bit file is the standard generated programming file. This is a binary configuration data file which contains header information that does not need to be downloaded to the FPGA. For generating the .bin file the BitGen option must be used. This is also a binary configuration data file but without header information. For configure the Kintex-7 FPGA of the TXMC638 both files could be used. Both binary configuration data file have addition data to the actual configuration data.

See also the XILINX User Guide (ug470) “7 Series FPGAs Configuration” for more information about Configuration Details and Configuration Data File Formats.

The following BitGen options are mandatory for the Slave Select Map Configuration via BBC.

- External Clock Master (53.2MHz) must be used.
- In contrast to SPI Configuration Mode, the Falling Edge Option must be switched off.

Additional important BitGen Options:

- For a faster configuration the Bitstream Compression could be used.
- The Persist Option is not needed. But if this option is used, the User FPGA must be set into reconfigure Mode by using the “FP_RE_CFG” Bit of the User FPGA Configuration Control/Status Register before Programming or Clearing the SPI Flash.

Xilinx Tandem Configuration Feature could not be used for Slave Select Map Configuration. It is therefore necessary to remove the Tandem Configuration Feature from the PCIe IP Core.

A design that is intended for the SPI configuration cannot be used by Slave Select Map configuration and vice versa.

7.4.4 Configuration via JTAG

The TXMC638 provides two JTAG chains which are accessible by one of the following connector options:

User JTAG Chain

- JST XRS Debug Connector

TEWS Factory configuration Chain

- XMC Connector P15

The User JTAG Chain is accessible from the JST XRS Debug Connector.

For direct FPGA configuration, FPGA read back or in-system diagnostics with Vivado Logic Analyzer, the JST XRS Debug Connector can be used to access the JTAG-chain. Also an indirect SPI – PROM programming is possible via JTAG Chain.

TEWS provides a “Programming Kit” (TA308) which includes a XSR cable and an adapter module that provides a Xilinx USB Programmer II compatible 2 mm shrouded header

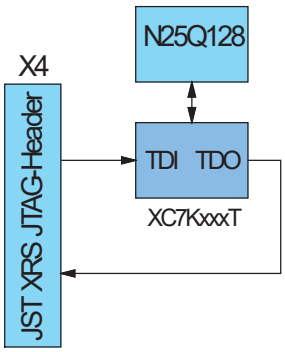


Figure 7-3 : User JTAG-Chain

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

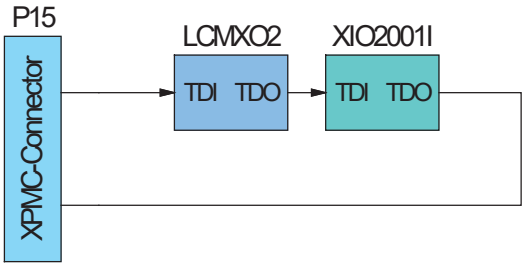


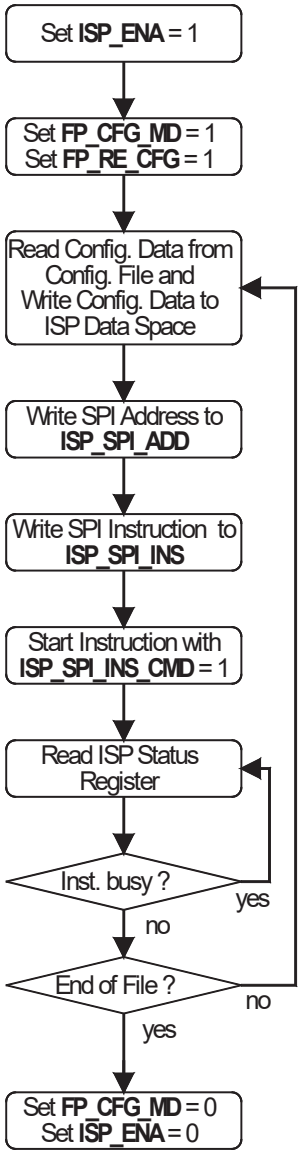
Figure 7-4 : TEWS Factory JTAG-Chain

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. If the FPGA is not configured or if it is possible that the FPGA accesses the SPI flash during BCC access set FP_RE_CFG = 0b1. Link must be set to disable previously!
- Read Configuration data from Configuration File and write Data to the In Circuit Programming Data Space. 256Byte (1 SPI Flash page) each time can be programmed maximum.
- Set the programming start address and write instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for next write instruction.
- Process should be repeated until all configuration data is written to the SPI Flash
- After completion the data programming, the ISP Mode bit must cleared to set configuration path to User FPGA and a Reconfiguration could be performed.



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

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

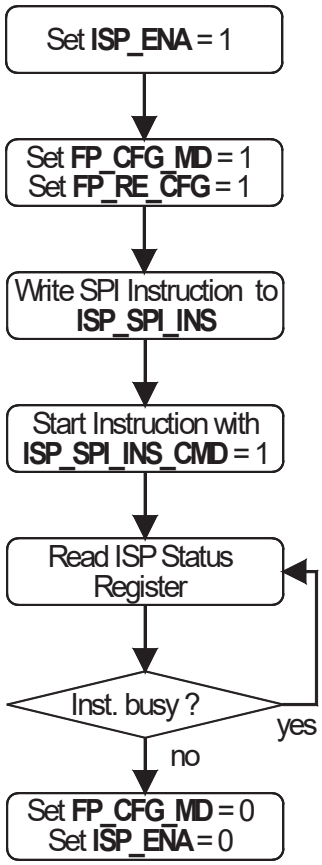
If not all configuration data bytes are written, the User FPGA is not configured correctly.

The source for the User FPGA SPI Configuration Flash data could be the .bin file. This file format can be created from the .bit file by using the XILINX Vivado software.

7.4.6 Erasing User FPGA SPI Configuration Flash

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

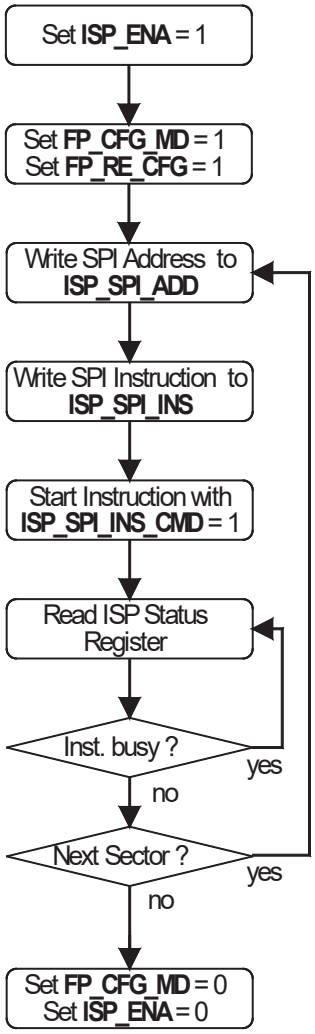
- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash. If the FPGA is not configured or if it is possible that the FPGA accesses the SPI flash during BCC access set FP_RE_CFG = 0b1. Link must be set to disable previously!
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for erasing process end.
- After completion of the erasing process, the ISP Mode bit should be cleared to set configuration path to User FPGA or a User FPGA SPI Configuration Flash programming process could be done.



7.4.7 Sector Erasing User FPGA SPI Configuration Flash

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

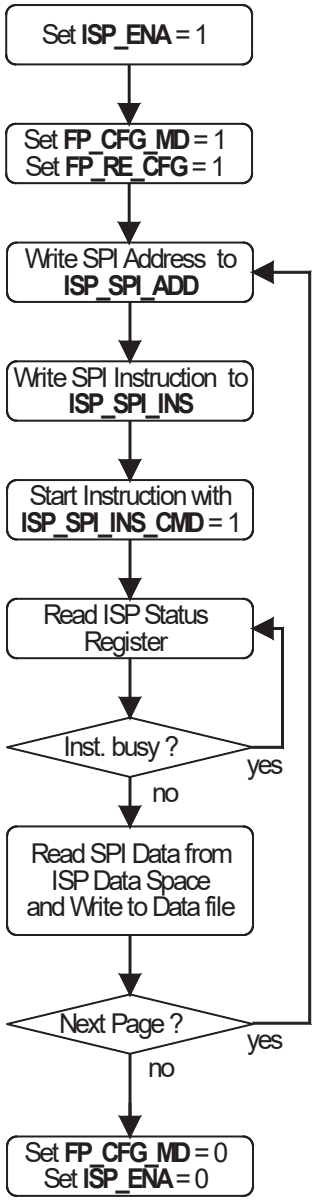
- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash. If the FPGA is not configured or if it is possible that the FPGA accesses the SPI flash during BCC access set FP_RE_CFG = 0b1. Link must be set to disable previously!
- Write the Sector Address to the ISP Configuration Register
- Set the Chip Erase instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for erasing process end.
- Process could be repeated for other sectors.
- After completion of the erasing process, the ISP Mode bit should be cleared to set configuration path to User FPGA or a User FPGA SPI Configuration Flash programming process could be done.



7.4.8 Reading User FPGA SPI Configuration Flash

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

- Enable the ISP Mode in the ISP Mode Enable Register.
- Assure that User FPGA Configuration Mode is set to SPI Flash. If the FPGA is not configured or if it is possible that the FPGA accesses the SPI flash during BCC access set FP_RE_CFG = 0b1. Link must be set to disable previously!
- Set the reading start address and write instruction in the ISP Configuration Register.
- Start the Instruction with ISP Command Register
- Wait on ISP SPI Instruction Done or ISP SPI Page Data Done for next write instruction.
- Read one page of SPI Data from In Circuit Programming Data Space and write to Data file
- Process could be repeated until all needed data are written to the Data file.
- After completion of the reading process, the ISP Mode bit must cleared to set configuration path back to User FPGA.



7.5 Board Configuration Controller (BCC – FPGA)

The Board Configuration FPGA is factory configured, and handles the basic board setup and User FPGA (Kintex-7) Configuration.

Changing or erase the BCF content leads to an inoperable TXMC638 FPGA configuration.

7.6 Clocking

7.6.1 FPGA Clock Sources

As a central clock generator of TXMC638 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.

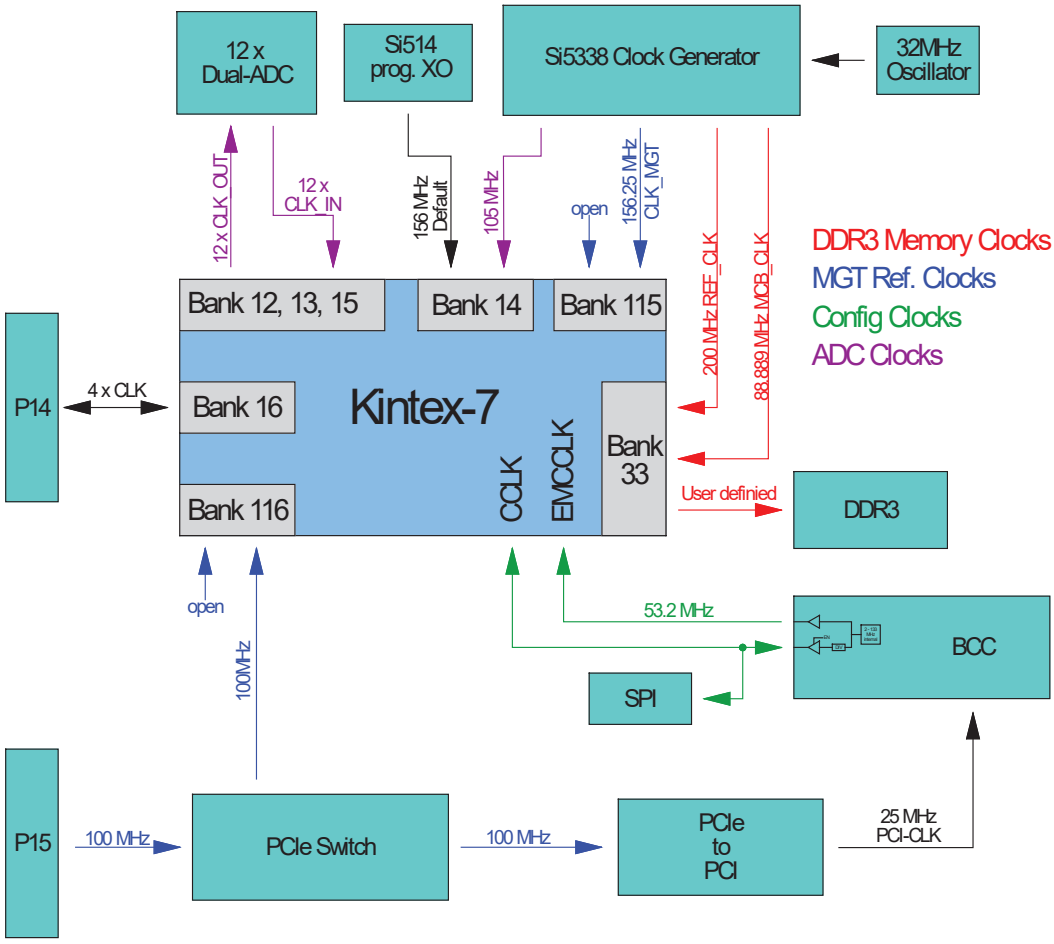


Figure 7-5 : FPGA Clock Sources

The following table lists the available clock sources on the TXMC638:

FPGA Clock-Pin Name	FPGA Pin Number	Source	Description
MGTREFCLK0_115	H6 / H5	SI5338 low-jitter clock generator	156.25 MHz differential MGT Reference clock
MGTREFCLK0_116	D6 / D5	PCIe Switch PI7C9X2G312GP	100 MHz differential PCIe Reference clock input
IO_L13P/N_T2_MRCC_33	AB11 / AC11	SI5338 low-jitter clock generator	88.889 MHz differential MCB CLK
IO_L14P/N_T2_SRCC_33	AA10 / AB10	SI5338 low-jitter clock generator	200 MHz differential Reference clock
IO_L12P_T1_MRCC_14	F22	SI5338 low-jitter clock generator	105 MHz Clock Input This clock is designated for ADC interface clock source.
IO_L13P/N_T2_MRCC_14	G22 / F23	Si514 prog. Oscillator	Differential free I2C prog. XO 100kHz up to 250MHz Default = 156 MHz
IO_L12P/N_T1_MRCC_12	Y23 / AA24	LTC2323	Diff. Clock ADC Ch. 1 and 2
IO_L13P/N_T2_MRCC_13	R21 / P21	LTC2323	Diff. Clock ADC Ch. 3 and 4
IO_L14P/N_T2_SRCC_12	AC23 / AC24	LTC2323	Diff. Clock ADC Ch. 5 and 6
IO_L14P/N_T2_SRCC_15	H17 / H18	LTC2323	Diff. Clock ADC Ch. 7 and 8
IO_L12P/N_T1_MRCC_15	F17 / E17	LTC2323	Diff. Clock ADC Ch. 9 and 10
IO_L12P/N_T1_MRCC_13	N21 / N22	LTC2323	Diff. Clock ADC Ch. 11 and 12
IO_L11P/N_T1_SRCC_12	AA23 / AB24	LTC2323	Diff. Clock ADC Ch. 13 and 14
IO_L14P/N_T2_SRCC_13	R22 / R23	LTC2323	Diff. Clock ADC Ch. 15 and 16
IO_L13P/N_T2_MRCC_12	Y22 / AA22	LTC2323	Diff. Clock ADC Ch. 17 and 18
IO_L11P/N_T1_SRCC_15	G17 / F18	LTC2323	Diff. Clock ADC Ch. 19 and 20
IO_L13P/N_T2_MRCC_15	E18 / D18	LTC2323	Diff. Clock ADC Ch. 21 and 22
IO_L11P/N_T1_SRCC_13	P23 / N23	LTC2323	Diff. Clock ADC Ch. 23 and 24
IO_L13P/N_T2_MRCC_16	C12 / C11	Back I/O Connector	Diff. P14 Back I/O Clock Input
IO_L14P/N_T2_SRCC_16	E11 / D11	Back I/O Connector	Diff. P14 Back I/O Clock Input
IO_L12P/N_T1_MRCC_16	E10 / D10	Back I/O Connector	Diff. P14 Back I/O Clock Input
IO_L11P/N_T1_SRCC_16	G11 / F10	Back I/O Connector	Diff. P14 Back I/O Clock Input
IO_L3N_T0_DQS_EMCCLK_14	B26	BCC	53.2 MHz used for external configuration clock (CCLK)

Table 7-5 : Available FPGA clocks

7.6.2 Si514 Free Programming Clock source

A second clock source is the free programmable Si514 XO from Silicon Laboratories.

The Si514 is programmable via I2C interface. An internal PLL allows output frequency of 100 kHz up to 250 MHz with programming resolution of 0.026 parts per billion.

The Si514 on TXMC638 is factory configured to 156 MHz default frequency.

The Si514 is connected via I2C interface to User FPGA (Kintex-7). As usual for the I2C interface the two pins must be realized as open drain buffer. The same I2C interface is used for the calibration data prom.

SPI-PROM Signal	Bank	V_{CCO}	Pin	Description / Kintex-7
FPGA_SCL	14	3.3V	F25	Serial clock
FPGA_SDA	14	3.3V	G26	Serial data

Table 7-6 : FPGA I2C SI514 Connections

Configuration and operation of the Si514 is controlled by reading and writing to the Si514 internal RAM space using the I2C interface. The device operates in slave mode with 7-bit addressing and can operate in Standard-Mode (100 kbps) or Fast-Mode (400 kbps).
The I2C 7-bit slave address of the Si514 is 0x55h.

For using the serial I2C interface between the USER FPGA (Kintex-7) and the Si514 please see also the Silicon Labs Si514 data sheet which describes the register map and serial communication process.

7.7 Back I/O Interface

P14 Back I/O Pins of the TXMC638 are direct routed to the User FPGA (Kintex-7). The I/O functions of these FPGA pins are directly dependent on the configuration of the FPGA.

The Kintex-7 VCCO voltage is set to 2.5V, so only the 2.5V I/O standards LVCMOS25, LVTTL25 and LVDS_25 are possible for using on TXMC638 back I/O interface.

Signal Name	Pin Number	Direction	IO Standard for example	IO Bank
BACK_IO0+	C12	IN/OUT	LVDS_25	16
BACK_IO0-	C11	IN/OUT	LVDS_25	16
BACK_IO1+	E11	IN/OUT	LVDS_25	16
BACK_IO1-	D11	IN/OUT	LVDS_25	16
BACK_IO2+	E10	IN/OUT	LVDS_25	16
BACK_IO2-	D10	IN/OUT	LVDS_25	16
BACK_IO3+	G11	IN/OUT	LVDS_25	16
BACK_IO3-	F10	IN/OUT	LVDS_25	16
BACK_IO4+	A9	IN/OUT	LVDS_25	16
BACK_IO4-	A8	IN/OUT	LVDS_25	16
BACK_IO5+	B10	IN/OUT	LVDS_25	16
BACK_IO5-	A10	IN/OUT	LVDS_25	16
BACK_IO6+	A13	IN/OUT	LVDS_25	16
BACK_IO6-	A12	IN/OUT	LVDS_25	16
BACK_IO7+	C9	IN/OUT	LVDS_25	16
BACK_IO7-	B9	IN/OUT	LVDS_25	16
BACK_IO8+	B14	IN/OUT	LVDS_25	16
BACK_IO8-	A14	IN/OUT	LVDS_25	16
BACK_IO9+	B12	IN/OUT	LVDS_25	16
BACK_IO9-	B11	IN/OUT	LVDS_25	16
BACK_IO10+	B15	IN/OUT	LVDS_25	16
BACK_IO10-	A15	IN/OUT	LVDS_25	16
BACK_IO11+	D9	IN/OUT	LVDS_25	16
BACK_IO11-	D8	IN/OUT	LVDS_25	16
BACK_IO12+	C14	IN/OUT	LVDS_25	16
BACK_IO12-	C13	IN/OUT	LVDS_25	16
BACK_IO13+	D14	IN/OUT	LVDS_25	16
BACK_IO13-	D13	IN/OUT	LVDS_25	16
BACK_IO14+	F9	IN/OUT	LVDS_25	16
BACK_IO14-	F8	IN/OUT	LVDS_25	16
BACK_IO15+	E13	IN/OUT	LVDS_25	16
BACK_IO15-	E12	IN/OUT	LVDS_25	16
BACK_IO16+	G10	IN/OUT	LVDS_25	16

BACK_IO16-	G9	IN/OUT	LVDS_25	16
BACK_IO17+	F14	IN/OUT	LVDS_25	16
BACK_IO17-	F13	IN/OUT	LVDS_25	16
BACK_IO18+	H9	IN/OUT	LVDS_25	16
BACK_IO18-	H8	IN/OUT	LVDS_25	16
BACK_IO19+	G12	IN/OUT	LVDS_25	16
BACK_IO19-	F12	IN/OUT	LVDS_25	16
BACK_IO20+	H12	IN/OUT	LVDS_25	16
BACK_IO20-	H11	IN/OUT	LVDS_25	16
BACK_IO21+	J11	IN/OUT	LVDS_25	16
BACK_IO21-	J10	IN/OUT	LVDS_25	16
BACK_IO22+	H14	IN/OUT	LVDS_25	16
BACK_IO22-	G14	IN/OUT	LVDS_25	16
BACK_IO23+	J13	IN/OUT	LVDS_25	16
BACK_IO23-	H13	IN/OUT	LVDS_25	16
BACK_IO24+	P16	IN/OUT	LVDS_25	13
BACK_IO24-	N17	IN/OUT	LVDS_25	13
BACK_IO25+	N18	IN/OUT	LVDS_25	13
BACK_IO25-	M19	IN/OUT	LVDS_25	13
BACK_IO26+	R16	IN/OUT	LVDS_25	13
BACK_IO26-	R17	IN/OUT	LVDS_25	13
BACK_IO27+	T19	IN/OUT	LVDS_25	13
BACK_IO27-	T18	IN/OUT	LVDS_25	13
BACK_IO28+	N19	IN/OUT	LVDS_25	13
BACK_IO28-	M20	IN/OUT	LVDS_25	13
BACK_IO29+	W20	IN/OUT	LVDS_25	12
BACK_IO29-	Y21	IN/OUT	LVDS_25	12
BACK_IO30+	R18	IN/OUT	LVDS_25	13
BACK_IO30-	P18	IN/OUT	LVDS_25	13
BACK_IO31+	U17	IN/OUT	LVDS_25	13
BACK_IO31-	T17	IN/OUT	LVDS_25	13

Table 7-7 : Digital Back I/O Interface

7.8 Memory

The TXMC638 is equipped with a 1 GB, 32 bit wide DDR3 SDRAM and a 128-Mbit non-volatile SPI-Flash. The SPI-Flash can also be used as configuration memory.

7.8.1 DDR3 SDRAM

The TXMC638 provides two MT41… (96-ball) DDR3 memory devices. The memory is accessible through a Memory Interface Controller Block IP in bank 32, 33 and 34 of the User FPGA.

Signal	DDR3 Pin	Termination	Memory Devices	
			Pin	Name
A0	AC8	49.9Ω VTT	N3	A0
A1	AA7	49.9Ω VTT	P7	A1
A2	AA8	49.9Ω VTT	P3	A2
A3	AF7	49.9Ω VTT	N2	A3
A4	AE7	49.9Ω VTT	P8	A4
A5	W8	49.9Ω VTT	P2	A5
A6	V9	49.9Ω VTT	R8	A6
A7	Y10	49.9Ω VTT	R2	A7
A8	Y11	49.9Ω VTT	T8	A8
A9	Y7	49.9Ω VTT	R3	A9
A10	Y8	49.9Ω VTT	L7	A10
A11	V7	49.9Ω VTT	R7	A11
A12	V8	49.9Ω VTT	N7	A12
A13	W11	49.9Ω VTT	T3	NC/A13
A14	V11	49.9Ω VTT	T7	NC/A14
BA0	AC7	49.9Ω VTT	M2	BA0
BA1	AB7	49.9Ω VTT	N8	BA1
BA2	AD8	49.9Ω VTT	M3	BA2
RAS#	AA9	49.9Ω VTT	J3	RAS#
CAS#	AB9	49.9Ω VTT	K3	CAS#
WE#	AC9	49.9Ω VTT	L3	WE#
RESET#	W10	4.7kΩ GND	T2	RESET#
CKE[0]	AB12	4.7kΩ GND	K9	CKE
ODT[0]	AC12	4.7kΩ GND	K1	ODT
DM_0	AE17	ODT	E7	LDM
DM_1	AA14	ODT	D3	UDM
DM_2	U6	ODT	E7	LDM
DM_3	Y3	ODT	D3	UDM
DQ0	AF17	ODT	E3	DQ0
DQ1	AF14	ODT	F7	DQ1
DQ2	AF15	ODT	F2	DQ2
DQ3	AD15	ODT	F8	DQ3

Signal	DDR3 Pin	Termination	Memory Devices	
			Pin	Name
DQ4	AE15	ODT	H3	DQ4
DQ5	AF19	ODT	H8	DQ5
DQ6	AF20	ODT	G2	DQ6
DQ7	AD16	ODT	H7	DQ7
DQ8	AA15	ODT	D7	DQ8
DQ9	AC14	ODT	C3	DQ9
DQ10	AD14	ODT	C8	DQ10
DQ11	AB14	ODT	C2	DQ11
DQ12	AB15	ODT	A7	DQ12
DQ13	AA17	ODT	A2	DQ13
DQ14	AA18	ODT	B8	DQ14
DQ15	AB16	ODT	A3	DQ15
DQ16	U5	ODT	E3	DQ0
DQ17	U2	ODT	F7	DQ1
DQ18	U1	ODT	F2	DQ2
DQ19	V3	ODT	F8	DQ3
DQ20	W3	ODT	H3	DQ4
DQ21	U7	ODT	H8	DQ5
DQ22	V6	ODT	G2	DQ6
DQ23	V4	ODT	H7	DQ7
DQ24	Y2	ODT	D7	DQ8
DQ25	V2	ODT	C3	DQ9
DQ26	V1	ODT	C8	DQ10
DQ27	W1	ODT	C2	DQ11
DQ28	Y1	ODT	A7	DQ12
DQ29	AB2	ODT	A2	DQ13
DQ30	AC2	ODT	B8	DQ14
DQ31	AA3	ODT	A3	DQ15
CK_p	AC13	100Ω	J7	CK
CK_n	AD13		K7	CK#
DQS_0_p	AE18	ODT	F3	LDQS
DQS_0_n	AF18	ODT	G3	LDQS#
DQS_1_p	Y15	ODT	C7	UDQS
DQS_1_n	Y16	ODT	B7	UDQS#
DQS_2_p	W6	ODT	F3	LDQS
DQS_2_n	W5	ODT	G3	LDQS#
DQS_3_p	AB1	ODT	C7	UDQS
DQS_3_n	AC1	ODT	B7	UDQS#

Table 7-8 : DDR3 SDRAM Interface

DDR3 Memory Device 01
DDR3 Memory Device 02
Both DDR3 Memory Devices 01 & 02

For details regarding the DDR3 SDRAM interface, please refer to XILINX Memory Interface Generator Documentation. Xilinx UG586: *Zynq-7000 AP SoC and 7 Series Devices Memory Interface Solutions v4.0*.

7.8.2 SPI-Flash

The TXMC638 provides a Micron N25Q128A 128-Mbit serial Flash memory. This Flash is used as FPGA configuration source (default configuration source).

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

The SPI-EEPROM is connected via Quad (x4) SPI interface to the User FPGA (Kintex-7) configuration interface.

SPI-PROM Signal	Bank	V_{CCO}	Pin	Description / Kintex-7
CLK	14	3.3V	C8	Serial Clock (CCLK_B)
CS#	14	3.3V	C23	Chip Select (FCS_B)
DI (bit0)	14	3.3V	B24	Serial Data input (MOSI) / MISO[0]
DO (bit1)	14	3.3V	A25	Serial Data output (DIN) / MISO[1]
WP# (bit2)	14	3.3V	B22	MISO[2] – D02
HOLD# (bit3)	14	3.3V	A22	MISO[3] – D03

Table 7-9 : FPGA SPI-Flash Connections

7.8.3 I2C - EEPROM

The TXMC638 provides an Atmel AT24C04D (512x8) I2C-Compatible (2-wire) Serial EEPROM.

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

The I2C EEPROM is connected via 2-wire interface to User FPGA (Kintex-7). As usual for the I2C interface the two pins must be realized as open drain buffer.

SPI-PROM Signal	Bank	V_{CCO}	Pin	Description / Kintex-7
FPGA_SCL	14	3.3V	F25	Serial clock
FPGA_SDA	14	3.3V	G26	Serial data

Table 7-10: FPGA I2C EEPROM Connections

For using the serial I2C interface between the USER FPGA (Kintex-7) and the I2C EEPROM please see the Atmel AT24C04D data sheet which describes the serial communication process.

7.8.3.1 I2C Calibration Data

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

ADC Offset Error:

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

ADC Gain Error:

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

All calibration data are determined with a common mode voltage of 0V DC. For a reliable correction for other common mode voltages, a new calibration of the customer environment is expedient.

7.8.3.2 ADC Calibration Data Values

The 24 ADC channels are realized with twelve dual LTC2323-16 ADC devices. For each device and for each channel respective correction values are stored.

I2C EEPROM Address	Description	Size (Bit)
0x000	ADC Channel 1 $Offset_{corr}$ High Byte	8
0x001	ADC Channel 1 $Offset_{corr}$ Low Byte	8
0x002	ADC Channel 1 $Gain_{corr}$ High Byte	8
0x003	ADC Channel 1 $Gain_{corr}$ Low Byte	8
0x004	ADC Channel 2 $Offset_{corr}$ High Byte	8
0x005	ADC Channel 2 $Offset_{corr}$ Low Byte	8
0x006	ADC Channel 2 $Gain_{corr}$ High Byte	8
0x007	ADC Channel 2 $Gain_{corr}$ Low Byte	8
0x008	ADC Channel 3 $Offset_{corr}$ High Byte	8
0x009	ADC Channel 3 $Offset_{corr}$ Low Byte	8
0x00A	ADC Channel 3 $Gain_{corr}$ High Byte	8
0x00B	ADC Channel 3 $Gain_{corr}$ Low Byte	8
0x00C	ADC Channel 4 $Offset_{corr}$ High Byte	8
0x00D	ADC Channel 4 $Offset_{corr}$ Low Byte	8
0x00E	ADC Channel 4 $Gain_{corr}$ High Byte	8
0x00F	ADC Channel 4 $Gain_{corr}$ Low Byte	8
…		
0x05C	ADC Channel 24 $Offset_{corr}$ High Byte	8
0x05D	ADC Channel 24 $Offset_{corr}$ Low Byte	8
0x05E	ADC Channel 24 $Gain_{corr}$ High Byte	8
0x05F	ADC Channel 24 $Gain_{corr}$ Low Byte	8

Table 7-11: ADC Calibration Data Values

7.8.3.3 ADC Data Correction Formula

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

The basic formula for correcting any ADC reading for the TXMC638 (bipolar input voltage range) is:

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

Value is the corrected result.

Reading is the data read from the ADC Data Register.

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

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

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

7.9 Serial ADC Interface

7.9.1 Overview

The 24 analog inputs of the TXMC638 are realized with 12 LTC2323-16 ADC devices. Each of these SAR-ADCs has two ADC channels. Thus, a total of 24 ADC channels are available on the TXMC638.

A coarse overview over the analog input section of the TXMC638 is shown by the following figure:

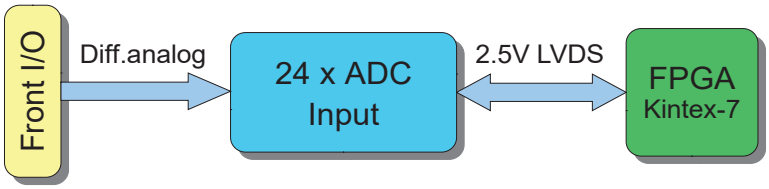


Figure 7-6 : Analog Input Section

The key-features of the LTC2323-16 are:

- Dual Channel - SAR-ADC
- 16 bit resolution
- 5Msps Throughput Rate for each Channel
- LVDS SPI-Compatible Serial Interface to User FPGA (Kintex-7)

In order to adapt the LTC2323-16 to a ±2.5V input voltage on each input-pin (±5V differential voltage range) and also the wide common mode voltage range of ±7.5 V two stage input operational amplifiers for input impedance conversion and gain adaption are needed in addition to the ADC.

The following figure shows the structure and principle of two ADC inputs. Both are connected via impedance converter and level adjustment to the LTC2313-16 dual ADC.

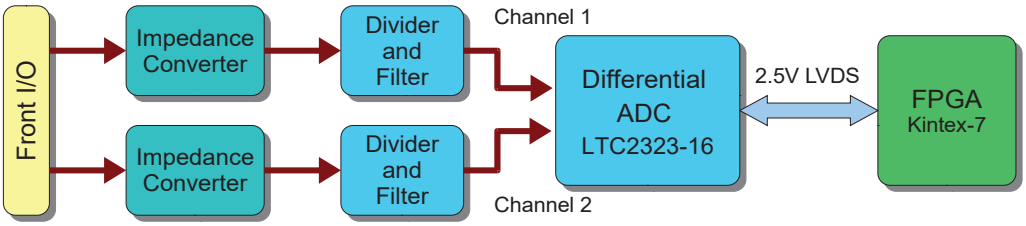


Figure 7-7 : Analog Input Block Diagram

7.9.2 ADC digital Output Coding

Differential common mode voltage is compensated by the analog input stage. In addition, the differential input voltage is reduced by an analogue divider which is built with a differential operational amplifier. The feedback resistors of this operational amplifier determine the divider value. The two resistance values 4k7 3k74 build together the divider of 1.2567. With maximum ±4.096 V analogue differential input voltage range of the ADC LTC2323-16 ensued in a theoretical maximum of ±5.1474 V differential input voltage range for a TXMC638 analog input channel.

Due to the ADC's true differential inputs, the ADC output coding significantly differs compared to a single ended input.

Analogue to a single ended input, where the range setting directly describes the ground related input voltage range, the ADC range setting describes the range of ground related voltages that can be tied to the ADC differential inputs. This results in an extended input voltage range, since the ADC measures the voltage between the differential inputs VIN- and VIN+.

An Example: The TXMC638 voltage range is rounded ±2.5 V, so the allowed (single ended, ground related) voltage on each ADC input pin is ±2.5 V. When we examine the two largest differential voltages, we get following results:

VIN-	VIN+	ADC Input Value
-2.5 V	+2.5 V	+5 V
+2.5 V	-2.5 V	-5 V

Table 7-12: ADC Data Coding Example

The example shows that the range of differential ADC input values is -5 V to +5 V, which results to a full scale range of 10 V for the ±2.5 V ADC Input Range. Similar

The TXMC638 data coding is two's complement.

Description	TXMC638	LTC2323-16	Digital Code
Full Scale Range	10.2948 V	8.192 V	-
Least Significant Bit	157 µV	125 µV	-
Full Scale (pos.)	5.147223 V	4.095875 V	0x7FFF
FSR - 1LSB	5.147066 V	4.095750 V	0x7FFE
Midscale + 1LSB	157 µV	125 V	0x0001
Midscale	0.0 V	0.0 V	0x0000
Midscale – 1LSB	-157 µV	-125 µV	0xFFFF
-FSR + 1LSB	-5.147223 V	-4.095875 V	0x8001
Full Scale (neg.)	-5.147380 V	-4.096000 V	0x8000

Table 7-13: ADC Data Coding

7.9.3 User FPGA Pinning

Each ADC is connected to the User FPGA (Kintex-7) via a dedicated serial clocked Interface. Each ADC device has one input clock, one output clock and one conversion signal. For each ADC channel there is a respective data output line, so both ADC channel transfers data at the same time.

Signal	Bank	VCCO	Pin	Description
SCK_00+	12	2.5V	AF24	Differential Clock Output for ADC Channel 1 and 2
SCK_00-	12	2.5V	AF25	
SCKOUT_00+	12	2.5V	Y23	Differential Clock Input for ADC Channel 1 and 2
SCKOUT_00-	12	2.5V	AA24	
SDO1_00+	12	2.5V	U24	Differential Data from ADC Channel 1
SDO1_00-	12	2.5V	U25	
SDO2_00+	12	2.5V	AD25	Differential Data from ADC Channel 2
SDO2_00-	12	2.5V	AE25	
CNV_N_00	12	2.5V	AB26	Convert Signal for ADC Channel 1 and 2

Signal	Bank	VCCO	Pin	Description
SCK_01+	13	2.5V	T24	Differential Clock Output for ADC Channel 3 and 4
SCK_01-	13	2.5V	T25	
SCKOUT_01+	13	2.5V	R21	Differential Clock Input for ADC Channel 3 and 4
SCKOUT_01-	13	2.5V	P21	
SDO1_01+	13	2.5V	T20	Differential Data from ADC Channel 3
SDO1_01-	13	2.5V	R20	
SDO2_01+	13	2.5V	T22	Differential Data from ADC Channel 4
SDO2_01-	13	2.5V	T23	
CNV_N_01	13	2.5V	P19	Convert Signal for ADC Channel 3 and 4

Signal	Bank	VCCO	Pin	Description
SCK_02+	12	2.5V	AB21	Differential Clock Output for ADC Channel 5 and 6
SCK_02-	12	2.5V	AC21	
SCKOUT_02+	12	2.5V	AC23	Differential Clock Input for ADC Channel 5 and 6
SCKOUT_02-	12	2.5V	AC24	
SDO1_02+	12	2.5V	V21	Differential Data from ADC Channel 5
SDO1_02-	12	2.5V	W21	
SDO2_02+	12	2.5V	AD21	Differential Data from ADC Channel 6
SDO2_02-	12	2.5V	AE21	
CNV_N_02	12	2.5V	Y26	Convert Signal for ADC Channel 5 and 6

Signal	Bank	VCCO	Pin	Description
SCK_03+	12	2.5V	AE22	Differential Clock Output for ADC Channel 7 and 8
SCK_03-	12	2.5V	AF22	
SCKOUT_03+	15	2.5V	H17	Differential Clock Input for ADC Channel 7 and 8
SCKOUT_03-	15	2.5V	H18	
SDO1_03+	15	2.5V	H16	Differential Data from ADC Channel 7
SDO1_03-	15	2.5V	G16	
SDO2_03+	15	2.5V	G19	Differential Data from ADC Channel 8
SDO2_03-	15	2.5V	F20	
CNV_N_03	12	2.5V	Y20	Convert Signal for ADC Channel 7 and 8

Signal	Bank	VCCO	Pin	Description
SCK_04+	12	2.5V	U26	Differential Clock Output for ADC Channel 9 and 10
SCK_04-	12	2.5V	V26	
SCKOUT_04+	15	2.5V	F17	Differential Clock Input for ADC Channel 9 and 10
SCKOUT_04-	15	2.5V	E17	
SDO1_04+	15	2.5V	C17	Differential Data from ADC Channel 9
SDO1_04-	15	2.5V	C18	
SDO2_04+	15	2.5V	C19	Differential Data from ADC Channel 10
SDO2_04-	15	2.5V	B19	
CNV_N_04	13	2.5V	M24	Convert Signal for ADC Channel 9 and 10

Signal	Bank	VCCO	Pin	Description
SCK_05+	13	2.5V	R25	Differential Clock Output for ADC Channel 11 and 12
SCK_05-	13	2.5V	P25	
SCKOUT_05+	13	2.5V	N21	Differential Clock Input for ADC Channel 11 and 12
SCKOUT_05-	13	2.5V	N22	
SDO1_05+	13	2.5V	N26	Differential Data from ADC Channel 11
SDO1_05-	13	2.5V	M26	
SDO2_05+	13	2.5V	R26	Differential Data from ADC Channel 12
SDO2_05-	13	2.5V	P26	
CNV_N_05	13	2.5V	P20	Convert Signal for ADC Channel 11 and 12

Signal	Bank	VCCO	Pin	Description
SCK_06+	12	2.5V	W23	Differential Clock Output for ADC Channel 13 and 14
SCK_06-	12	2.5V	W24	
SCKOUT_06+	12	2.5V	AA23	Differential Clock Input for ADC Channel 13 and 14
SCKOUT_06-	12	2.5V	AB24	
SDO1_06+	12	2.5V	V23	Differential Data from ADC Channel 13
SDO1_06-	12	2.5V	V24	
SDO2_06+	12	2.5V	U22	Differential Data from ADC Channel 14
SDO2_06-	12	2.5V	V22	
CNV_N_06	12	2.5V	Y25	Convert Signal for ADC Channel 13 and 14

Signal	Bank	VCCO	Pin	Description
SCK_07+	13	2.5V	U19	Differential Clock Output for ADC Channel 15 and 16
SCK_07-	13	2.5V	U20	
SCKOUT_07+	13	2.5V	R22	Differential Clock Input for ADC Channel 15 and 16
SCKOUT_07-	13	2.5V	R23	
SDO1_07+	12	2.5V	AD26	Differential Data from ADC Channel 15
SDO1_07-	12	2.5V	AE26	
SDO2_07+	13	2.5V	P24	Differential Data from ADC Channel 16
SDO2_07-	13	2.5V	N24	
CNV_N_07	13	2.5V	M22	Convert Signal for ADC Channel 15 and 16

Signal	Bank	VCCO	Pin	Description
SCK_08+	12	2.5V	AE23	Differential Clock Output for ADC Channel 17 and 18
SCK_08-	12	2.5V	AF23	
SCKOUT_08+	12	2.5V	Y22	Differential Clock Input for ADC Channel 17 and 18
SCKOUT_08-	12	2.5V	AA22	
SDO1_08+	12	2.5V	AB22	Differential Data from ADC Channel 17
SDO1_08-	12	2.5V	AC22	
SDO2_08+	12	2.5V	AD23	Differential Data from ADC Channel 18
SDO2_08-	12	2.5V	AD24	
CNV_N_08	12	2.5V	AC26	Convert Signal for ADC Channel 17 and 18

Signal	Bank	VCCO	Pin	Description
SCK_09+	12	2.5V	AA25	Differential Clock Output for ADC Channel 19 and 20
SCK_09-	12	2.5V	AB25	
SCKOUT_09+	15	2.5V	G17	Differential Clock Input for ADC Channel 19 and 20
SCKOUT_09-	15	2.5V	F18	
SDO1_09+	15	2.5V	F19	Differential Data from ADC Channel 19
SDO1_09-	15	2.5V	E20	
SDO2_09+	15	2.5V	H19	Differential Data from ADC Channel 20
SDO2_09-	15	2.5V	G20	
CNV_N_09	13	2.5V	U16	Convert Signal for ADC Channel 19 and 20

Signal	Bank	VCCO	Pin	Description
SCK_10+	12	2.5V	W25	Differential Clock Output for ADC Channel 21 and 22
SCK_10-	12	2.5V	W26	
SCKOUT_10+	15	2.5V	E18	Differential Clock Input for ADC Channel 21 and 22
SCKOUT_10-	15	2.5V	D18	
SDO1_10+	15	2.5V	A18	Differential Data from ADC Channel 21
SDO1_10-	15	2.5V	A19	
SDO2_10+	15	2.5V	C16	Differential Data from ADC Channel 22
SDO2_10-	15	2.5V	B16	
CNV_N_10	13	2.5V	L24	Convert Signal for ADC Channel 21 and 22

Signal	Bank	VCCO	Pin	Description
SCK_11+	13	2.5V	K25	Differential Clock Output for ADC Channel 23 and 24
SCK_11-	13	2.5V	K26	
SCKOUT_11-	13	2.5V	N23	Differential Clock Input for ADC Channel 23 and 24
SCKOUT_11+	13	2.5V	P23	
SDO1_11+	15	2.5V	K20	Differential Data from ADC Channel 23
SDO1_11-	15	2.5V	J20	
SDO2_11+	13	2.5V	M25	Differential Data from ADC Channel 24
SDO2_11-	13	2.5V	L25	
CNV_N_11	13	2.5V	M21	Convert Signal for ADC Channel 23 and 24

Table 7-14: ADC Interface Connections

For using the clocked serial interface between the User FPGA (Kintex-7) and one of the twelve LTC2323-16 ADC devices please use the LTC2323-16 data sheet which describes the communication process.

7.9.4 Programming Hints LTC2323-16

The LTC2323-16 digital interface is a simple clocked SPI based interface.

This differential interface uses differential LVDS signals for serial data transfer. All LVDS signals need a termination on the receiver side of the connection. For the SCK± an external resistor is implemented on the TMXC638. The User FPGA inputs CLKOUT±, SDO1± and SDO2± of each ADC channel need an FPGA internal termination. The corresponding constrains for the pin assignment, the I/O standard, termination and slew rate is specified in chapter 11 Appendix A.

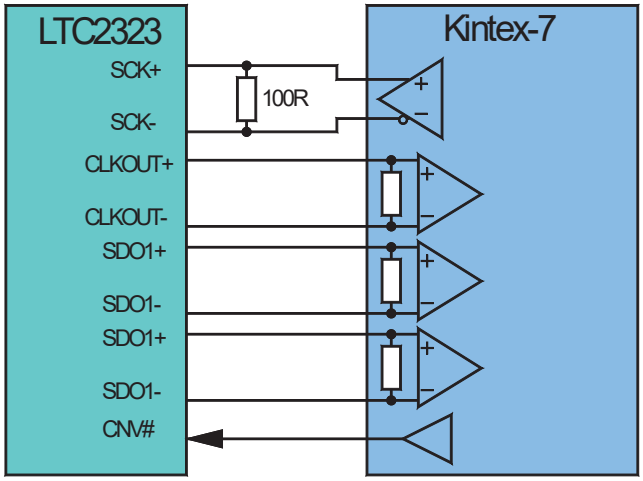


Figure 7-8 : Digital ADC to FPGA Interface

A conversion is triggered by a negative edge on the CNV# line. The acquisition is done during the positive phase of the CNV# signal. Following the FPGA drives the SCK clock, which then initiates the data transfer from the ADC to the FPGA. The ADC then transmits the serial data SDO1 / SDO2 synchronous to CLKOUT. The data sequence is MSB first and the LSB at least.

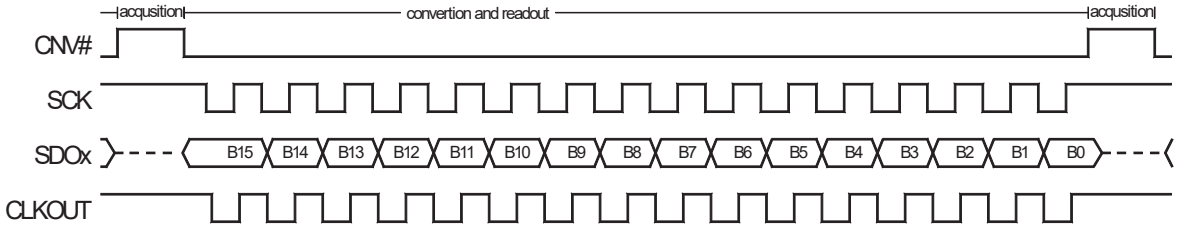


Figure 7-9 : Timing Diagram LTC2323-16

Note, that the one-cycle conversion latency has the result that the previous sample word is transmitted first. That means, at the beginning of a burst sampling period the first conversion result will be invalid.

A detailed description of the LTC2323-16 interface and the LTC2323-16 function please use the data sheet which describes the whole communication process and all special characteristics of the ADC.

7.10 AC coupled differential Inputs

The TXMC638 provides three 100 Ohm terminated, ac-coupled, differential Inputs.
These inputs could be used as GPIO inputs or trigger input for ADC conversion. Despite the ac-coupling, also dc-signals are supported after an initial edge.

The differential input is converted into an LVDS signal and connected to User FPGA (Kintex-7).

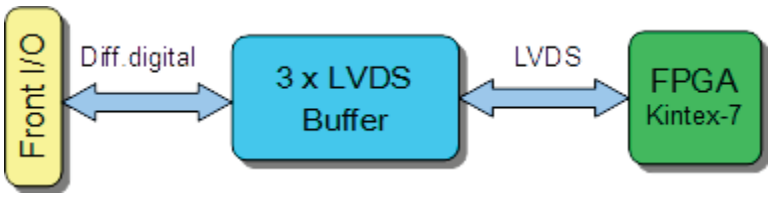


Figure 7-10: Block Diagram differential Inputs

These three User FPGA (Kintex-7) differential inputs always need a differential Termination. LVDS Input 2 does not need due to the 1.35V VCCO bank I/O supply an internal termination. The both other LVDS Inputs need an FPGA internal termination. The corresponding constrains for the pin assignment, the I/O standard, termination and slew rate is specified in chapter 11 Appendix A.

Signal	Bank	VCCO	Pin	Termination	Description
DIFF_K7_0+	15	2.5V	L19	needs internal diff. Termination	Differential LVDS Input 0 use LVDS25 I/O Standard
DIFF_K7_0-	15	2.5V	L20		
DIFF_K7_1+	15	2.5V	J18	needs internal diff. Termination	Differential LVDS Input 1 use LVDS25 I/O Standard
DIFF_K7_1-	15	2.5V	J19		
DIFF_K7_2+	34	1.35V	AA4	external Termination is provided	Differential LVDS Input 2 use LVDS I/O Standard
DIFF_K7_2-	34	1.35V	AB4		

Table 7-15: AC coupled differential Inputs

7.11 Serial Number Allocation

The TXMC638 Module Serial Number is stored on-board the module, and can be read on both FPGA devices. The BCC provides a Serial Number Register in the local register space.

For the User FPGA (Kintex-7) an I2C Master interface is required to read the serial number from the Configuration FPGA (BCC). For this purpose the Configuration FPGA (BCC) provides an I2C slave interface.

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

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

The Configuration I2C Interface provides only one readable register. The Serial Number Register is a 32 bit wide read only register. The Slave Address of the Serial Number Register is 0b0110101.

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 TXMC638 module	R	-

Table 7-17: TXMC638 Serial Number

Example: 0x0091_981A => SNo.: 9541658

7.11.1 Device Addressing and Operation

The TXMC638 Configuration FPGA uses a standard 7 bit Slave Address. The eighth bit of the slave address is the Read/write operation select bit.

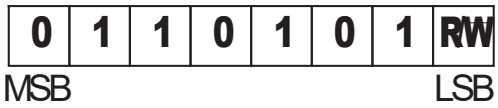


Figure 7-11: Configuration FPGA Slave Address

TXMC638 Configuration FPFA I2C Slave typically Start and Stop condition

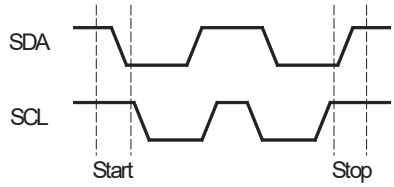


Figure 7-12: Configuration FPGA Start and Stop condition

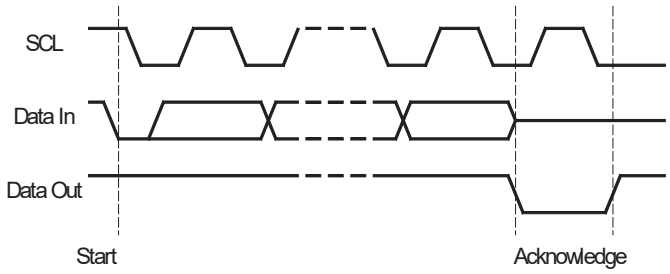


Figure 7-13: Configuration FPGA Output Acknowledge

7.11.2 Read Operation

The TXMC638 Configuration FPGA provides only one 32 bit register which could be read from User FPGA via a I2C Interface.

The read operation starts with a I2C start condition followed by a 7 bit slave address. The read/write bit in the device address byte is set to one. The configuration FPGA acknowledged the address and began to transmit all four data byte of the TXM638 Serial Number Register. Each byte must be acknowledged. The sequence must be completed with a stop condition by the User FPGA.

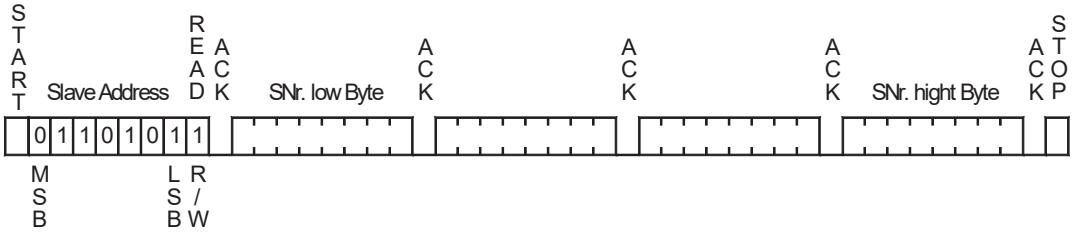


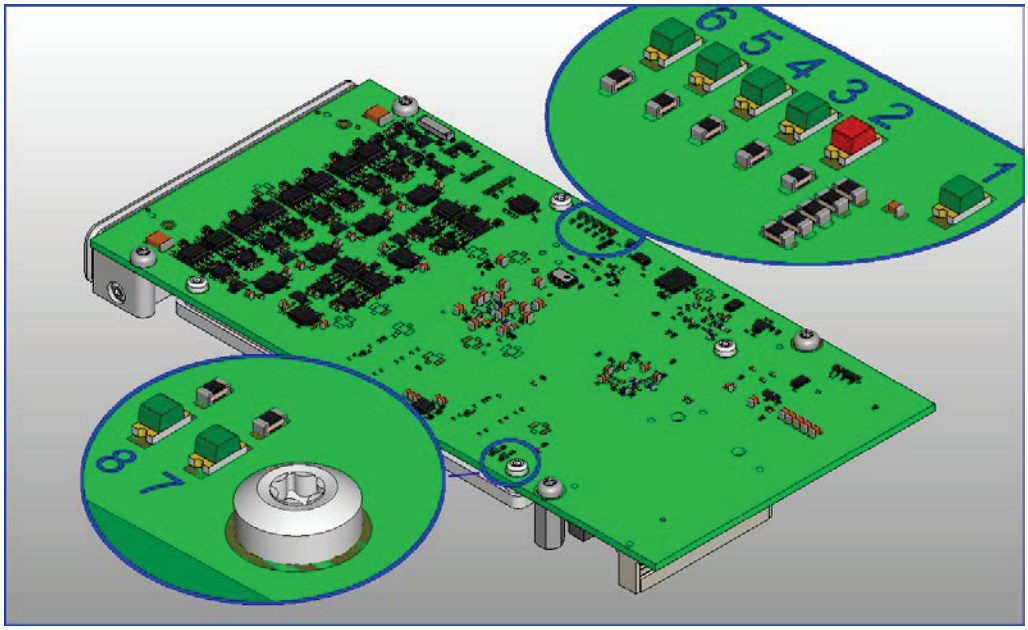
Figure 7-14: Configuration FPGA Slave Access

7.11.3 Write Operation

A I2C write operation is not implemented.

7.12 On-Board Indicators

The TXMC638 provides a couple of board-status LEDs as shown below. These include Power-Good and FPGA configuration status indications as well as two general purpose LEDs.



No.	LED	Color	State		Description
1	Power Good	Green	off	On-Board Power Supplies are not ok	Power Good Signal for all on-board power supplies.
			on	On-Board Power Supplies are all ok	
2	User INIT	Red	off	INIT state is inactive	User FPGA (Kintex-7) INIT# - Pin LED. Indicates FPGA configuration
			on	INIT state is active	
3	User DONE	Green	off	Device is not configured	User FPGA (Kintex-7) DONE-Pin LED. Indicates successful FPGA configuration
			on	Device is completely configured	
4	BCC DONE	Green	off	Device is not configured	Board Configuration FPGA DONE-Pin LED Indicates successful FPGA configuration and initiation.
			flashing	Device is completely configured	
			on	Device is completely configured	
5	GPIO LED1	Green	off	General State is not IDLE	Configuration FPGA depends.
			on	General State is IDLE	
6	GPIO LED2	Green	off	PCI Reset is active	
			on	PCI Reset is inactive	
7	USER LED1	Green	-		Design dependent, can be controlled by the User FPGA. Refer to chapter “User-GPIO”
8	USER LED2	Green	-		

Table 7-18: Board-Status and User LEDs

7.13 Thermal Management

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

To avoid permanent damage of the User FPGA, the temperature of the Kintex-7 should be monitored.

All components on the TXMC638 are rated for the industrial temperature range of -40°C to +85°C. The module has several hot spots (input operational amplifier, ADC devices, power supplies) but the main hot-spot is the User FPGA (Kintex-7). The heat sink is directly mounted to the Kintex-7 die. For maximum heat transfer the heat sink is embodied so large that it covers a lot of the module.

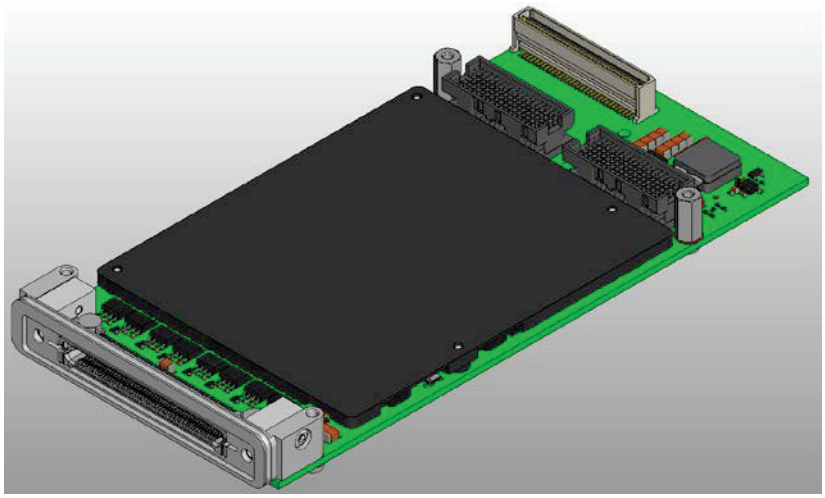


Figure 7-15: TXMC638 with Heatsink

The actual achievable ambient operating temperature range is highly dependent on the FPGA design, the load of the modules I/O circuitry and the applied cooling method.

A simple system air cooling is not sufficient heavy utilization of FPGA. Since the heat sink of the TXMC638 always mounted between PCB of the Carrier and PCB of the TXMC, targeted ventilation is highly recommended.

Use the Xilinx XPower Estimator (XPE) or XPower Analyzer to determine whether additional cooling requirements such as forced air cooling apply. It is also strongly recommend to use the internal temperature monitoring of the Kintex-7, or via the XADC to monitor the temperature. Corresponding descriptions can be found in the documentation XILINX.

8 Design Help

8.1 Board Reference Design

User applications for the TXMC638 may be developed by using the TXMC638 FPGA Board Reference Design.

TEWS offers this Board Reference Design as a well-documented example basic example. It includes an .xdc constrain file with all necessary pin assignments and basic timing constraints. The example design covers the main functionalities of the TXMC638. It implements a PCIe endpoint with interrupt support, register mapping, DDR3 memory access and basic I/O functions. It comes as a Xilinx Vivado 2016.1 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 TXMC638 FPGA Application design can be developed using the design software Vivado Design Suite. Licenses for design tools are required.

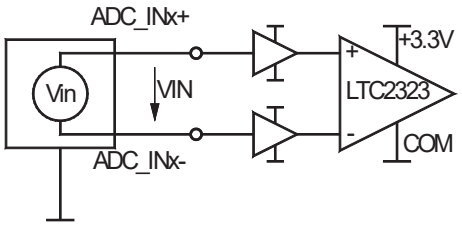
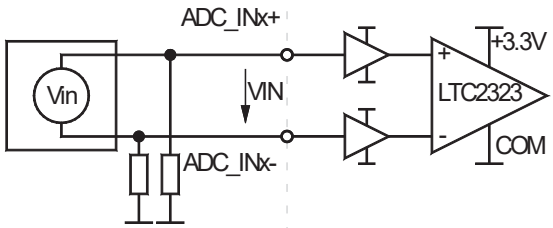
For TXMC638 FPGA Example Application design see also the included User Manual.

9 Installation

9.1 I/O Interface

9.1.1 Front I/O - ADC Analog Input Level

All analog inputs are connected via an impedance converter and a second operation amplifier for level adjustment and filtering to the differential ADC inputs. This also serves as a protection of the ADC from excessive analog input levels.

Diff. Source with Ground Reference	Diff. Source without Ground Reference
	
Maximum VIN = ±5.0V	Maximum VIN = ±5.0V

The TXMC638 has differential analog inputs. When talking about the input voltage range of a differential input, one has to differentiate between the differential input voltage between the two pins, and the input voltage relative to ground for each pin.

With an input voltage range of ±2.5V (ground related) for each pin of the differential input, we get the ±5.0V differential input voltages:

Voltage	Description	TXMC638
V_{diff_max}	Maximum differential input voltage (range)	±5.0 V
V_{cm_max}	Maximum common mode input voltage (range)	±7.5 V
V_{in_max}	Operating voltage range of the input pins	±10.0 V
V_{in_abs}	Absolute maximum voltage range of the input	Power Off: ±5.0 V Power On: ±18.5 V

Table 9-1 : Differential Input Voltage Ranges

The range of this differential input is -5V to +5V, which results to a full scale range of 10V for the ±2.5V (per pin, relative to ground) Input voltage range.

Additional statement

A differential input voltage of ±5V means that the difference between the two input pins VIN+ and VIN- of the differential input can range from +5V to -5V, but contains no information about the allowed voltage relative to ground for each input pin. If e.g. VIN+ = 98V and VIN- = 102V, the differential input voltage would be -4V. But the input common mode voltage is 100V, truly damaging the TXMC638.

Example analog Input Level:

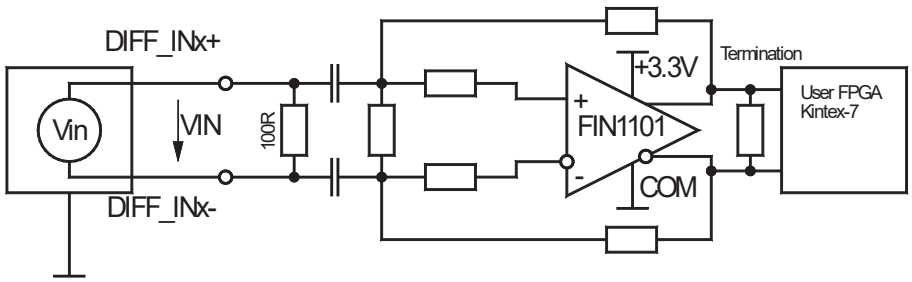
At the vertical black dashed line

VIN+	=	**-10 V**					
VIN-	=	**-5 V**					
VDIFF	=	-10	-	-5 V		=	**-5 V**
VCM	=	-10	+	-5 V	/2	=	**-7.5 V**

9.1.2 Front I/O – AC coupled differential Inputs

All three AC coupled differential inputs are connected via an high speed single port LVDS buffer from the front I/O connector to the User FPGA (Kintex-7). The wide input voltage range provides several applications like external clock input, trigger input or simple digital input.

Differential Source with Ground Reference



VIN = ±200mV up to ±3.6V

9.1.3 Back I/O Interface

P14 Back I/O Pins of the TXMC638 are direct routed to the User FPGA (Kintex-7). The I/O functions of these FPGA pins are directly dependent on the configuration of the FPGA.

The Kintex-7 VCCO voltage is set to 2.5V, so only the 2.5V I/O standards LVCMOS25 and LVDS_25 are possible for using on TXMC638 back I/O interface.

9.2 FPGA JTAG Connector

The FPGA JTAG connector X4 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 Vivado Logic Analyzer).

The Debug Connector provides the User FPGA (Kintex-7) JTAG interface and two TXMC638 status signals.

- The JTAG interface consists of the signals TDI, TDO, TMS, TCK, uses 3.3V I/O voltage, and can run with up to 10 MHz.
- The first status signal indicates the state of the supply voltage (status Power GOOD). The second status signal of the FPGA JTAG connector indicates the configuration state of the User FPGA (status DONE).

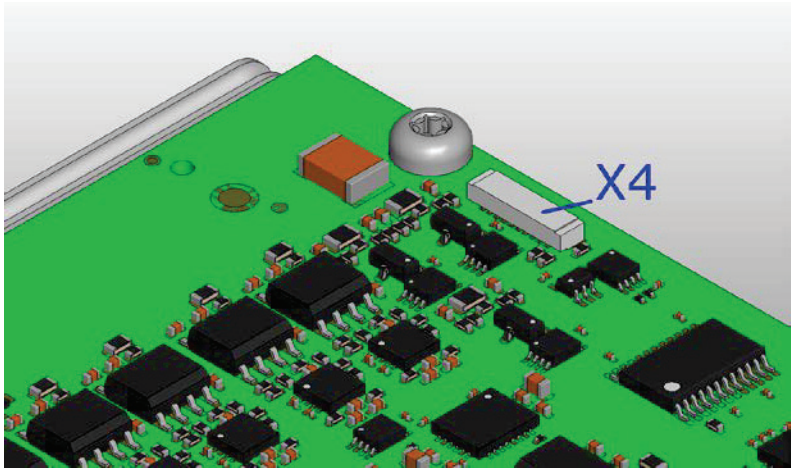
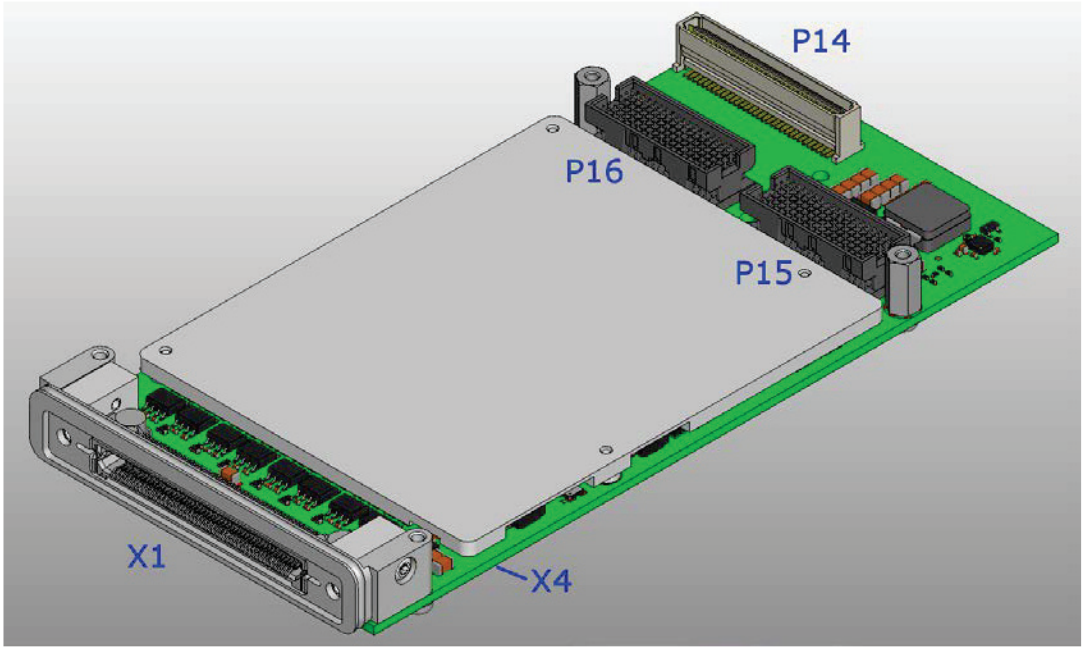


Figure 9-1 : FPGA JTAG Connector X4

TEWS provides a "Programming Kit" (TA308) which includes a XSR cable and an adapter module that provides a Xilinx USB Programmer II compatible 2 mm shrouded header.

10 Pin Assignment – I/O Connector

10.1 Overview

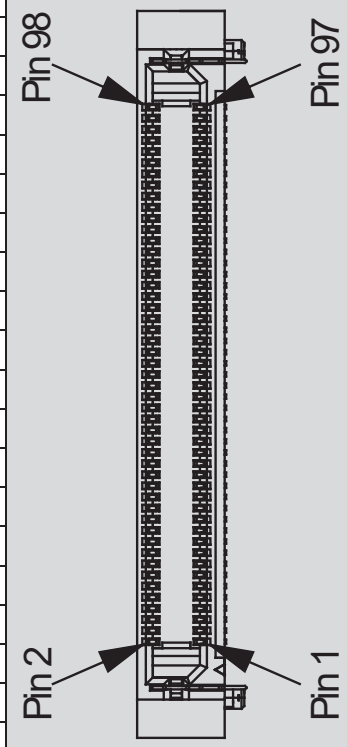


10.2 X1 Front Panel I/O Connector

10.2.1 Connector Type

Pin-Count	98
Connector Type	Rugged EdgeRate female connector
Source & Order Info	Samtec – ERF8-049-01-L-D-RA-L

10.2.2 Pin Assignment

Pin	I/O	Connector View	Pin	I/O
1	GND		2	GND
3	ADC_IN1+		4	ADC_IN13+
5	ADC_IN1-		6	ADC_IN13-
7	GND		8	GND
9	ADC_IN2+		10	ADC_IN14+
11	ADC_IN2-		12	ADC_IN14-
13	GND		14	GND
15	ADC_IN3+		16	ADC_IN15+
17	ADC_IN3-		18	ADC_IN15-
19	GND		20	GND
21	ADC_IN4+		22	ADC_IN16+
23	ADC_IN4-		24	ADC_IN16-
25	GND		26	GND
27	ADC_IN5+		28	ADC_IN17+
29	ADC_IN5-		30	ADC_IN17-
31	GND		32	GND
33	ADC_IN6+		34	ADC_IN18+
35	ADC_IN6-		36	ADC_IN18-
37	GND		38	GND
39	ADC_IN7+		40	ADC_IN19+
41	ADC_IN7-		42	ADC_IN19-
43	GND		44	GND
45	ADC_IN8+		46	ADC_IN20+
47	ADC_IN8-		48	ADC_IN20-
49	GND		50	GND
51	ADC_IN9+		52	ADC_IN21+
53	ADC_IN9-		54	ADC_IN21-
55	GND		56	GND
57	ADC_IN10+		58	ADC_IN22+
59	ADC_IN10-		60	ADC_IN22-
61	GND		62	GND
63	ADC_IN11+		64	ADC_IN23+
65	ADC_IN11-		66	ADC_IN23-

Pin	I/O	Connector View	Pin	I/O
67	GND		68	GND
69	ADC_IN12+		70	ADC_IN24+
71	ADC_IN12-		72	ADC_IN24-
73	GND		74	GND
75	n.c.		76	n.c.
77	n.c.		78	n.c.
79	GND		80	GND
81	n.c.		82	DIFF_IN2+
83	n.c.		84	DIFF_IN2-
85	GND		86	GND
87	n.c.		88	DIFF_IN1+
89	n.c.		90	DIFF_IN1-
91	GND		92	GND
93	n.c.		94	DIFF_IN0+
95	n.c.		96	DIFF_IN0-
97	GND		98	GND

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-

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

Figure 10-1: Pin Assignment P14 Back I/O Connector TXMC638

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-	-	Tx 3+	Tx 3-	-
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-	-	Rx 3+	Rx 3-	-
14	GND	GND	-	GND	GND	-
15	-	-	-	-	-	-
16	GND	GND	-	GND	GND	-
17	-	-	-	-	-	-
18	GND	GND	-	GND	GND	-
19	-	-	-	-	-	-

Figure 10-2: Pin Assignment P16 Back I/O Connector TXMC638

10.5 X4 FPGA 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 TEWS TA308 Cable Kit. In conjunction with this Cable Kit, the Xilinx Platform Cable USB II could be connected to the TXMC638. This allows the direct usage of Xilinx software-tools like Vivado Logic Analyzer or the Vivado Hardware Manager.

10.5.1 Connector Type

Pin-Count	10
Connector Type	JST XRS 10pol 0,6 mm Pitch IDC Connector
Source & Order Info	SM10B-XSRS-ETB
Mating Part	10XSR-36S (Cable)

10.5.2 Pin Assignment

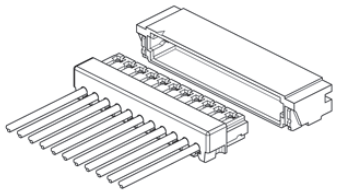
	Pin	Description
	1	GND
	2	TCK
	3	TMS
	4	TDI
	5	TDO
	6	GND
	7	GPIO0 (DONE)
	8	GPIO1 (PGOOD)
	9	n.c.
	10	V_{REF}

Table 10-2: Pin Assignment FPGA JTAG Header X4

TEWS provides a "Programming Kit" (TA308) which includes a XSR cable and an adapter module that provides a Xilinx USB Programmer II compatible 2 mm shrouded header.

11 Appendix A

This appendix contains the signal to pin assignments for the User FPGA Kintex-7.

```
## ############################################################################################## ##
##                                       TEWS TECHNOLOGIES                                      ##
## ############################################################################################## ##
##
##   Project Name    : TXMC638 Example
##   File Name       : txmc638_exa.xdc
##   Target Device   : XC7K160T-FBG676-1
##   Design Tool     : Xilinx Vivado Design Suite Design Edition 2015.3
##   Simulation Tool :
##
##   Description     : Constraint file TXMC638 FPGA/K7 Firmware
##
##   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) 2016
##                     TEWS TECHNOLOGIES GmbH
##
##   History         :
##       Version   1 : (SE, 07.06.2016)
##                     Initial Version
##       Version   2 : (SE, 15.06.2016)
##                     - Added master SPI x4 configuration mode
##                     - Added clock constraints for ADC bit clocks (adc_clkout_x)
##                     - Added timing constraints for ADC data inputs (adc_sdo1/2_x)
##       Version   3 : (SE, 17.06.2016)
##                     - Added external master clock usage w/o divider
##                     - Added differential termination for ADC data inputs (adc_sdo1/2_x)
##       Version   4 : (SE, 24.06.2016)
##                     - Set SPI falling edge alignment (to improve SPI timing)
##                       - Set asynchronous clock group constraints for ADC bit clock (adc_bclk_x) to 105MHz clock
domain (USER_CLKA)
##       Version   5 : (SE, 04.07.2016)
##                              - Removed SPI falling edge alignment (to improve SPI timing) to allow SelectMAP
configuration
##       Version   6 : (SE, 11.07.2016)
##                          - Added dual purpose I/O persistence (BITSTREAM.CONFIG.PERSIST) and configuration mode
(CONFIG_MODE)
##                        - Added data path timing constraints for ADC bit clock (adc_bclk_x) from/to 105MHz clock
domain (USER_CLKA)
##       Version   7 : (SE, 24.08.2016)
##                                  - Changed I/O locations due to new PCB version (CNV_3/4/9/10, DIFF_K7_0/1,
SCK_0/3/4/8/9/10, SCKOUT_0, SDO1_7/11, SDO2_0/2/5)
##                     - Changed iostandard for I/Os DIFF_K7_0/1 incl. internal termination (DIFF_TERM TRUE)
##       Version   8 : (SE, 26.08.2016)
##                     - Added SPI falling edge constraint (SPI_FALL_EDGE)
##                     - Added bitstream compression (COMPRESS)
##
##   Comments        : none#
## ############################################################################################## ##


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

# Bitstream Setting
set_property BITSTREAM.CONFIG.EXTMASTERCCLK_EN div-1 [current_design]
set_property BITSTREAM.CONFIG.PERSIST YES [current_design]

set_property CONFIG_MODE SPIx4 [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 4 [current_design]

#SE, 28.08.2016: Falling edge setting is required for direct used external master clock (EXTMASTERCCLK_EN div-1)
set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]

set_property BITSTREAM.GENERAL.COMPRESS true [current_design]


## ############################################################################################## ##
## Section: MGT
```

```
## ############################################################################################ ##

# XMC Lanes
#set_property                          LOC                          GTXE2_CHANNEL_X0Y4                          [get_cells
{B_PCIe_FW_UNIT.I_PCIe_FW_UNIT/B_PCIE_EP.I_PCIE_EP/U0/inst/gt_top_i/pipe_wrapper_i/pipe_lane[3].gt_wrapper_i/gtx_c
hannel.gtxe2_channel_i}]
set_property PACKAGE_PIN F2 [get_ports {PER_P[3]}]
set_property PACKAGE_PIN F1 [get_ports {PER_N[3]}]
set_property PACKAGE_PIN G4 [get_ports {PET_P[3]}]
set_property PACKAGE_PIN G3 [get_ports {PET_N[3]}]

#set_property                          LOC                          GTXE2_CHANNEL_X0Y5                          [get_cells
{B_PCIe_FW_UNIT.I_PCIe_FW_UNIT/B_PCIE_EP.I_PCIE_EP/U0/inst/gt_top_i/pipe_wrapper_i/pipe_lane[2].gt_wrapper_i/gtx_c
hannel.gtxe2_channel_i}]
set_property PACKAGE_PIN D2 [get_ports {PER_P[2]}]
set_property PACKAGE_PIN D1 [get_ports {PER_N[2]}]
set_property PACKAGE_PIN E4 [get_ports {PET_P[2]}]
set_property PACKAGE_PIN E3 [get_ports {PET_N[2]}]

#set_property                          LOC                          GTXE2_CHANNEL_X0Y6                          [get_cells
{B_PCIe_FW_UNIT.I_PCIe_FW_UNIT/B_PCIE_EP.I_PCIE_EP/U0/inst/gt_top_i/pipe_wrapper_i/pipe_lane[1].gt_wrapper_i/gtx_c
hannel.gtxe2_channel_i}]
set_property PACKAGE_PIN B2 [get_ports {PER_P[1]}]
set_property PACKAGE_PIN B1 [get_ports {PER_N[1]}]
set_property PACKAGE_PIN C4 [get_ports {PET_P[1]}]
set_property PACKAGE_PIN C3 [get_ports {PET_N[1]}]

#set_property                          LOC                          GTXE2_CHANNEL_X0Y7                          [get_cells
{B_PCIe_FW_UNIT.I_PCIe_FW_UNIT/B_PCIE_EP.I_PCIE_EP/U0/inst/gt_top_i/pipe_wrapper_i/pipe_lane[0].gt_wrapper_i/gtx_c
hannel.gtxe2_channel_i}]
set_property PACKAGE_PIN A4 [get_ports {PER_P[0]}]
set_property PACKAGE_PIN A3 [get_ports {PER_N[0]}]
set_property PACKAGE_PIN B6 [get_ports {PET_P[0]}]
set_property PACKAGE_PIN B5 [get_ports {PET_N[0]}]


# XMC Reference Clock
set_property PACKAGE_PIN D6 [get_ports REFCLK_02_P]
set_property PACKAGE_PIN D5 [get_ports REFCLK_02_N]

create_clock -period 10.000 [get_ports REFCLK_02_P]


## ############################################################################################ ##
## Section: BCC
## ############################################################################################ ##

#set_property PACKAGE_PIN L23 [get_ports FPGA_RST_n]
#set_property IOSTANDARD LVCMOS33 [get_ports FPGA_RST_n]


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

set_property IOSTANDARD LVCMOS33 [get_ports USER_CLKA]
set_property PACKAGE_PIN F22 [get_ports USER_CLKA]

create_clock -period 9.5238 [get_ports USER_CLKA]
set_clock_groups -asynchronous -group {USER_CLKA} -group {userclk1}


#set_property SLEW FAST [get_ports {Si514_CLK_P}]
#set_property IOSTANDARD LVDS_25 [get_ports {Si514_CLK_P}]  # External Termination
#set_property PACKAGE_PIN M25 [get_ports {Si514_CLK_P}]
#
#set_property SLEW FAST [get_ports {Si514_CLK_N}]
#set_property IOSTANDARD LVDS_25 [get_ports {Si514_CLK_N}]  # External Termination
#set_property PACKAGE_PIN L25 [get_ports {Si514_CLK_N}]


## ############################################################################################ ##
## Section: PCIe Switch
## ############################################################################################ ##

set_property IOSTANDARD LVCMOS33 [get_ports DWN_RST_n]
set_property PACKAGE_PIN K21 [get_ports DWN_RST_n]


## ############################################################################################ ##
## Section: DDR3
## ############################################################################################ ##
```

```
# DDR3 Data (DQ)
set_property SLEW FAST [get_ports {DQ[0]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[0]}]
set_property PACKAGE_PIN AF17 [get_ports {DQ[0]}]

set_property SLEW FAST [get_ports {DQ[1]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[1]}]
set_property PACKAGE_PIN AF14 [get_ports {DQ[1]}]

set_property SLEW FAST [get_ports {DQ[2]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[2]}]
set_property PACKAGE_PIN AF15 [get_ports {DQ[2]}]

set_property SLEW FAST [get_ports {DQ[3]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[3]}]
set_property PACKAGE_PIN AD15 [get_ports {DQ[3]}]

set_property SLEW FAST [get_ports {DQ[4]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[4]}]
set_property PACKAGE_PIN AE15 [get_ports {DQ[4]}]

set_property SLEW FAST [get_ports {DQ[5]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[5]}]
set_property PACKAGE_PIN AF19 [get_ports {DQ[5]}]

set_property SLEW FAST [get_ports {DQ[6]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[6]}]
set_property PACKAGE_PIN AF20 [get_ports {DQ[6]}]

set_property SLEW FAST [get_ports {DQ[7]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[7]}]
set_property PACKAGE_PIN AD16 [get_ports {DQ[7]}]

set_property SLEW FAST [get_ports {DQ[8]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[8]}]
set_property PACKAGE_PIN AA15 [get_ports {DQ[8]}]

set_property SLEW FAST [get_ports {DQ[9]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[9]}]
set_property PACKAGE_PIN AC14 [get_ports {DQ[9]}]

set_property SLEW FAST [get_ports {DQ[10]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[10]}]
set_property PACKAGE_PIN AD14 [get_ports {DQ[10]}]

set_property SLEW FAST [get_ports {DQ[11]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[11]}]
set_property PACKAGE_PIN AB14 [get_ports {DQ[11]}]

set_property SLEW FAST [get_ports {DQ[12]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[12]}]
set_property PACKAGE_PIN AB15 [get_ports {DQ[12]}]

set_property SLEW FAST [get_ports {DQ[13]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[13]}]
set_property PACKAGE_PIN AA17 [get_ports {DQ[13]}]

set_property SLEW FAST [get_ports {DQ[14]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[14]}]
set_property PACKAGE_PIN AA18 [get_ports {DQ[14]}]

set_property SLEW FAST [get_ports {DQ[15]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[15]}]
set_property PACKAGE_PIN AB16 [get_ports {DQ[15]}]

set_property SLEW FAST [get_ports {DQ[16]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[16]}]
set_property PACKAGE_PIN U5 [get_ports {DQ[16]}]

set_property SLEW FAST [get_ports {DQ[17]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[17]}]
set_property PACKAGE_PIN U2 [get_ports {DQ[17]}]

set_property SLEW FAST [get_ports {DQ[18]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[18]}]
set_property PACKAGE_PIN U1 [get_ports {DQ[18]}]

set_property SLEW FAST [get_ports {DQ[19]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[19]}]
set_property PACKAGE_PIN V3 [get_ports {DQ[19]}]

set_property SLEW FAST [get_ports {DQ[20]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[20]}]
```

```
set_property PACKAGE_PIN W3 [get_ports {DQ[20]}]

set_property SLEW FAST [get_ports {DQ[21]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[21]}]
set_property PACKAGE_PIN U7 [get_ports {DQ[21]}]

set_property SLEW FAST [get_ports {DQ[22]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[22]}]
set_property PACKAGE_PIN V6 [get_ports {DQ[22]}]

set_property SLEW FAST [get_ports {DQ[23]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[23]}]
set_property PACKAGE_PIN V4 [get_ports {DQ[23]}]

set_property SLEW FAST [get_ports {DQ[24]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[24]}]
set_property PACKAGE_PIN Y2 [get_ports {DQ[24]}]

set_property SLEW FAST [get_ports {DQ[25]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[25]}]
set_property PACKAGE_PIN V2 [get_ports {DQ[25]}]

set_property SLEW FAST [get_ports {DQ[26]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[26]}]
set_property PACKAGE_PIN V1 [get_ports {DQ[26]}]

set_property SLEW FAST [get_ports {DQ[27]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[27]}]
set_property PACKAGE_PIN W1 [get_ports {DQ[27]}]

set_property SLEW FAST [get_ports {DQ[28]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[28]}]
set_property PACKAGE_PIN Y1 [get_ports {DQ[28]}]

set_property SLEW FAST [get_ports {DQ[29]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[29]}]
set_property PACKAGE_PIN AB2 [get_ports {DQ[29]}]

set_property SLEW FAST [get_ports {DQ[30]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[30]}]
set_property PACKAGE_PIN AC2 [get_ports {DQ[30]}]

set_property SLEW FAST [get_ports {DQ[31]}]
set_property IOSTANDARD SSTL135_T_DCI [get_ports {DQ[31]}]
set_property PACKAGE_PIN AA3 [get_ports {DQ[31]}]


# DDR3 Address
set_property SLEW FAST [get_ports {A[0]}]
set_property IOSTANDARD SSTL135 [get_ports {A[0]}]
set_property PACKAGE_PIN AC8 [get_ports {A[0]}]

set_property SLEW FAST [get_ports {A[1]}]
set_property IOSTANDARD SSTL135 [get_ports {A[1]}]
set_property PACKAGE_PIN AA7 [get_ports {A[1]}]

set_property SLEW FAST [get_ports {A[2]}]
set_property IOSTANDARD SSTL135 [get_ports {A[2]}]
set_property PACKAGE_PIN AA8 [get_ports {A[2]}]

set_property SLEW FAST [get_ports {A[3]}]
set_property IOSTANDARD SSTL135 [get_ports {A[3]}]
set_property PACKAGE_PIN AF7 [get_ports {A[3]}]

set_property SLEW FAST [get_ports {A[4]}]
set_property IOSTANDARD SSTL135 [get_ports {A[4]}]
set_property PACKAGE_PIN AE7 [get_ports {A[4]}]

set_property SLEW FAST [get_ports {A[5]}]
set_property IOSTANDARD SSTL135 [get_ports {A[5]}]
set_property PACKAGE_PIN W8 [get_ports {A[5]}]

set_property SLEW FAST [get_ports {A[6]}]
set_property IOSTANDARD SSTL135 [get_ports {A[6]}]
set_property PACKAGE_PIN V9 [get_ports {A[6]}]

set_property SLEW FAST [get_ports {A[7]}]
set_property IOSTANDARD SSTL135 [get_ports {A[7]}]
set_property PACKAGE_PIN Y10 [get_ports {A[7]}]

set_property SLEW FAST [get_ports {A[8]}]
set_property IOSTANDARD SSTL135 [get_ports {A[8]}]
set_property PACKAGE_PIN Y11 [get_ports {A[8]}]
```

```
set_property SLEW FAST [get_ports {A[9]}]
set_property IOSTANDARD SSTL135 [get_ports {A[9]}]
set_property PACKAGE_PIN Y7 [get_ports {A[9]}]

set_property SLEW FAST [get_ports {A[10]}]
set_property IOSTANDARD SSTL135 [get_ports {A[10]}]
set_property PACKAGE_PIN Y8 [get_ports {A[10]}]

set_property SLEW FAST [get_ports {A[11]}]
set_property IOSTANDARD SSTL135 [get_ports {A[11]}]
set_property PACKAGE_PIN V7 [get_ports {A[11]}]

set_property SLEW FAST [get_ports {A[12]}]
set_property IOSTANDARD SSTL135 [get_ports {A[12]}]
set_property PACKAGE_PIN V8 [get_ports {A[12]}]

set_property SLEW FAST [get_ports {A[13]}]
set_property IOSTANDARD SSTL135 [get_ports {A[13]}]
set_property PACKAGE_PIN W11 [get_ports {A[13]}]

set_property SLEW FAST [get_ports {A[14]}]
set_property IOSTANDARD SSTL135 [get_ports {A[14]}]
set_property PACKAGE_PIN V11 [get_ports {A[14]}]


# DDR3 Bank Address (BA)
set_property SLEW FAST [get_ports {BA[0]}]
set_property IOSTANDARD SSTL135 [get_ports {BA[0]}]
set_property PACKAGE_PIN AC7 [get_ports {BA[0]}]

set_property SLEW FAST [get_ports {BA[1]}]
set_property IOSTANDARD SSTL135 [get_ports {BA[1]}]
set_property PACKAGE_PIN AB7 [get_ports {BA[1]}]

set_property SLEW FAST [get_ports {BA[2]}]
set_property IOSTANDARD SSTL135 [get_ports {BA[2]}]
set_property PACKAGE_PIN AD8 [get_ports {BA[2]}]


# DDR3 Row Address Strobe (RAS)
set_property SLEW FAST [get_ports {RAS_n}]
set_property IOSTANDARD SSTL135 [get_ports {RAS_n}]
set_property PACKAGE_PIN AA9 [get_ports {RAS_n}]


# DDR3 Column Address Strobe (RAS)
set_property SLEW FAST [get_ports {CAS_n}]
set_property IOSTANDARD SSTL135 [get_ports {CAS_n}]
set_property PACKAGE_PIN AB9 [get_ports {CAS_n}]


# DDR3 Write Enable (WE_n, active-low)
set_property SLEW FAST [get_ports {WE_n}]
set_property IOSTANDARD SSTL135 [get_ports {WE_n}]
set_property PACKAGE_PIN AC9 [get_ports {WE_n}]


# DDR3 Reset (RST_n, active-low)
set_property SLEW FAST [get_ports {RST_n}]
set_property IOSTANDARD SSTL135 [get_ports {RST_n}]
set_property PACKAGE_PIN W10 [get_ports {RST_n}]


# DDR3 Clock Enable (CKE)
set_property SLEW FAST [get_ports {CKE[0]}]
set_property IOSTANDARD SSTL135 [get_ports {CKE[0]}]
set_property PACKAGE_PIN AB12 [get_ports {CKE[0]}]


# DDR3 On-Die Termination (ODT)
set_property SLEW FAST [get_ports {ODT[0]}]
set_property IOSTANDARD SSTL135 [get_ports {ODT[0]}]
set_property PACKAGE_PIN AC12 [get_ports {ODT[0]}]


# DDR3 Chip Select (CS, active-low)
set_property SLEW FAST [get_ports {CS_n[0]}]
set_property IOSTANDARD SSTL135 [get_ports {CS_n[0]}]
set_property PACKAGE_PIN AA13 [get_ports {CS_n[0]}]


# DDR3 Data Mask (DM)
```

```
set_property SLEW FAST [get_ports {DM[0]}]
set_property IOSTANDARD SSTL135 [get_ports {DM[0]}]
set_property PACKAGE_PIN AE17 [get_ports {DM[0]}]

set_property SLEW FAST [get_ports {DM[1]}]
set_property IOSTANDARD SSTL135 [get_ports {DM[1]}]
set_property PACKAGE_PIN AA14 [get_ports {DM[1]}]

set_property SLEW FAST [get_ports {DM[2]}]
set_property IOSTANDARD SSTL135 [get_ports {DM[2]}]
set_property PACKAGE_PIN U6 [get_ports {DM[2]}]

set_property SLEW FAST [get_ports {DM[3]}]
set_property IOSTANDARD SSTL135 [get_ports {DM[3]}]
set_property PACKAGE_PIN Y3 [get_ports {DM[3]}]


# DDR3 Data Strobes (DQS)
set_property SLEW FAST [get_ports {DQS_P[0]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_P[0]}]
set_property PACKAGE_PIN AE18 [get_ports {DQS_P[0]}]

set_property SLEW FAST [get_ports {DQS_N[0]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_N[0]}]
set_property PACKAGE_PIN AF18 [get_ports {DQS_N[0]}]

set_property SLEW FAST [get_ports {DQS_P[1]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_P[1]}]
set_property PACKAGE_PIN Y15 [get_ports {DQS_P[1]}]

set_property SLEW FAST [get_ports {DQS_N[1]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_N[1]}]
set_property PACKAGE_PIN Y16 [get_ports {DQS_N[1]}]

set_property SLEW FAST [get_ports {DQS_P[2]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_P[2]}]
set_property PACKAGE_PIN W6 [get_ports {DQS_P[2]}]

set_property SLEW FAST [get_ports {DQS_N[2]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_N[2]}]
set_property PACKAGE_PIN W5 [get_ports {DQS_N[2]}]

set_property SLEW FAST [get_ports {DQS_P[3]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_P[3]}]
set_property PACKAGE_PIN AB1 [get_ports {DQS_P[3]}]

set_property SLEW FAST [get_ports {DQS_N[3]}]
set_property IOSTANDARD DIFF_SSTL135_T_DCI [get_ports {DQS_N[3]}]
set_property PACKAGE_PIN AC1 [get_ports {DQS_N[3]}]


# DDR3 System Clock
set_property SLEW FAST [get_ports {CK_P[0]}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {CK_P[0]}]
set_property PACKAGE_PIN AC13 [get_ports {CK_P[0]}]

set_property SLEW FAST [get_ports {CK_N[0]}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {CK_N[0]}]
set_property PACKAGE_PIN AD13 [get_ports {CK_N[0]}]

# DDR3 Memory Clock
set_property SLEW FAST [get_ports {MCB_CLK_P}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {MCB_CLK_P}]
set_property PACKAGE_PIN AB11 [get_ports {MCB_CLK_P}]

set_property SLEW FAST [get_ports {MCB_CLK_N}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {MCB_CLK_N}]
set_property PACKAGE_PIN AC11 [get_ports {MCB_CLK_N}]

create_clock -period 11.25 [get_ports MCB_CLK_P]


# DDR3 Reference Clock
set_property SLEW FAST [get_ports {REF_CLK_P}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {REF_CLK_P}]
set_property PACKAGE_PIN AA10 [get_ports {REF_CLK_P}]

set_property SLEW FAST [get_ports {REF_CLK_N}]
set_property IOSTANDARD DIFF_SSTL135 [get_ports {REF_CLK_N}]
set_property PACKAGE_PIN AB10 [get_ports {REF_CLK_N}]

create_clock -period 5 [get_ports REF_CLK_P]
```

```
## ############################################################################################ ##
## Section: ADCs (LTC2323)
## ############################################################################################ ##

# ADC #0
set_property SLEW FAST [get_ports {ADC_CNV_n[0]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[0]}]
set_property PACKAGE_PIN AB26 [get_ports {ADC_CNV_n[0]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[0]}]
# External Termination
set_property PACKAGE_PIN AF24 [get_ports {ADC_SCK_P[0]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[0]}]
# External Termination
set_property PACKAGE_PIN AF25 [get_ports {ADC_SCK_N[0]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[0]}]
set_property PACKAGE_PIN Y23 [get_ports {ADC_SCKOUT_P[0]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[0]}]
set_property PACKAGE_PIN AA24 [get_ports {ADC_SCKOUT_N[0]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[0]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[0]}]
set_property PACKAGE_PIN U24 [get_ports {ADC_SDO1_P[0]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[0]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[0]}]
set_property PACKAGE_PIN U25 [get_ports {ADC_SDO1_N[0]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[0]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[0]}]
set_property PACKAGE_PIN AD25 [get_ports {ADC_SDO2_P[0]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[0]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[0]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[0]}]
set_property PACKAGE_PIN AE25 [get_ports {ADC_SDO2_N[0]}]


# Timings
create_clock -name adc_bclk_0 -period 9.523809523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[0]}]

set_input_delay -clock adc_bclk_0 -min 0 [get_ports {ADC_SDO1_P[0]}]
set_input_delay -clock adc_bclk_0 -max 2 [get_ports {ADC_SDO1_P[0]}]
set_input_delay -clock adc_bclk_0 -min 0 [get_ports {ADC_SDO2_P[0]}]
set_input_delay -clock adc_bclk_0 -max 2 [get_ports {ADC_SDO2_P[0]}]

set_clock_groups -asynchronous -group {adc_bclk_0} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_0} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_0};


# ADC #1
set_property SLEW FAST [get_ports {ADC_CNV_n[1]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[1]}]
set_property PACKAGE_PIN P19 [get_ports {ADC_CNV_n[1]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[1]}]
# External Termination
set_property PACKAGE_PIN T24 [get_ports {ADC_SCK_P[1]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[1]}]
# External Termination
set_property PACKAGE_PIN T25 [get_ports {ADC_SCK_N[1]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[1]}]
set_property PACKAGE_PIN R21 [get_ports {ADC_SCKOUT_P[1]}]
```

```
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[1]}]
set_property PACKAGE_PIN P21 [get_ports {ADC_SCKOUT_N[1]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[1]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[1]}]
set_property PACKAGE_PIN T20 [get_ports {ADC_SDO1_P[1]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[1]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[1]}]
set_property PACKAGE_PIN R20 [get_ports {ADC_SDO1_N[1]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[1]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[1]}]
set_property PACKAGE_PIN T22 [get_ports {ADC_SDO2_P[1]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[1]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[1]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[1]}]
set_property PACKAGE_PIN T23 [get_ports {ADC_SDO2_N[1]}]


# Timings
create_clock -name adc_bclk_1 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[1]}]

set_input_delay -clock adc_bclk_1 -min 0 [get_ports {ADC_SDO1_P[1]}]
set_input_delay -clock adc_bclk_1 -max 2 [get_ports {ADC_SDO1_P[1]}]
set_input_delay -clock adc_bclk_1 -min 0 [get_ports {ADC_SDO2_P[1]}]
set_input_delay -clock adc_bclk_1 -max 2 [get_ports {ADC_SDO2_P[1]}]

set_clock_groups -asynchronous -group {adc_bclk_1} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_1} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_1};


# ADC #2
set_property SLEW FAST [get_ports {ADC_CNV_n[2]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[2]}]
set_property PACKAGE_PIN Y26 [get_ports {ADC_CNV_n[2]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[2]}]
# External Termination
set_property PACKAGE_PIN AB21 [get_ports {ADC_SCK_P[2]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[2]}]
# External Termination
set_property PACKAGE_PIN AC21 [get_ports {ADC_SCK_N[2]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[2]}]
set_property PACKAGE_PIN AC23 [get_ports {ADC_SCKOUT_P[2]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[2]}]
set_property PACKAGE_PIN AC24 [get_ports {ADC_SCKOUT_N[2]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[2]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[2]}]
set_property PACKAGE_PIN V21 [get_ports {ADC_SDO1_P[2]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[2]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[2]}]
set_property PACKAGE_PIN W21 [get_ports {ADC_SDO1_N[2]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[2]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[2]}]
set_property PACKAGE_PIN AD21 [get_ports {ADC_SDO2_P[2]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[2]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[2]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[2]}]
set_property PACKAGE_PIN AE21 [get_ports {ADC_SDO2_N[2]}]
```

```
# Timings
create_clock -name adc_bclk_2 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[2]}]

set_input_delay -clock adc_bclk_2 -min 0 [get_ports {ADC_SDO1_P[2]}]
set_input_delay -clock adc_bclk_2 -max 2 [get_ports {ADC_SDO1_P[2]}]
set_input_delay -clock adc_bclk_2 -min 0 [get_ports {ADC_SDO2_P[2]}]
set_input_delay -clock adc_bclk_2 -max 2 [get_ports {ADC_SDO2_P[2]}]

set_clock_groups -asynchronous -group {adc_bclk_2} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_2} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_2};


# ADC #3
set_property SLEW FAST [get_ports {ADC_CNV_n[3]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[3]}]
set_property PACKAGE_PIN Y20 [get_ports {ADC_CNV_n[3]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[3]}]
# External Termination
set_property PACKAGE_PIN AE22 [get_ports {ADC_SCK_P[3]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[3]}]
# External Termination
set_property PACKAGE_PIN AF22 [get_ports {ADC_SCK_N[3]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[3]}]
set_property PACKAGE_PIN H17 [get_ports {ADC_SCKOUT_P[3]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[3]}]
set_property PACKAGE_PIN H18 [get_ports {ADC_SCKOUT_N[3]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[3]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[3]}]
set_property PACKAGE_PIN H16 [get_ports {ADC_SDO1_P[3]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[3]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[3]}]
set_property PACKAGE_PIN G16 [get_ports {ADC_SDO1_N[3]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[3]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[3]}]
set_property PACKAGE_PIN G19 [get_ports {ADC_SDO2_P[3]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[3]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[3]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[3]}]
set_property PACKAGE_PIN F20 [get_ports {ADC_SDO2_N[3]}]


# Timings
create_clock -name adc_bclk_3 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[3]}]

set_input_delay -clock adc_bclk_3 -min 0 [get_ports {ADC_SDO1_P[3]}]
set_input_delay -clock adc_bclk_3 -max 2 [get_ports {ADC_SDO1_P[3]}]
set_input_delay -clock adc_bclk_3 -min 0 [get_ports {ADC_SDO2_P[3]}]
set_input_delay -clock adc_bclk_3 -max 2 [get_ports {ADC_SDO2_P[3]}]

set_clock_groups -asynchronous -group {adc_bclk_3} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_3} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_3};


# ADC #4
set_property SLEW FAST [get_ports {ADC_CNV_n[4]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[4]}]
set_property PACKAGE_PIN M24 [get_ports {ADC_CNV_n[4]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[4]}]
# External Termination
set_property PACKAGE_PIN U26 [get_ports {ADC_SCK_P[4]}]
```

```
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[4]}]
# External Termination
set_property PACKAGE_PIN V26 [get_ports {ADC_SCK_N[4]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[4]}]
set_property PACKAGE_PIN F17 [get_ports {ADC_SCKOUT_P[4]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[4]}]
set_property PACKAGE_PIN E17 [get_ports {ADC_SCKOUT_N[4]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[4]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[4]}]
set_property PACKAGE_PIN C17 [get_ports {ADC_SDO1_P[4]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[4]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[4]}]
set_property PACKAGE_PIN C18 [get_ports {ADC_SDO1_N[4]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[4]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[4]}]
set_property PACKAGE_PIN C19 [get_ports {ADC_SDO2_P[4]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[4]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[4]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[4]}]
set_property PACKAGE_PIN B19 [get_ports {ADC_SDO2_N[4]}]


# Timings
create_clock -name adc_bclk_4 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[4]}]

set_input_delay -clock adc_bclk_4 -min 0 [get_ports {ADC_SDO1_P[4]}]
set_input_delay -clock adc_bclk_4 -max 2 [get_ports {ADC_SDO2_P[4]}]
set_input_delay -clock adc_bclk_4 -min 0 [get_ports {ADC_SDO1_P[4]}]
set_input_delay -clock adc_bclk_4 -max 2 [get_ports {ADC_SDO2_P[4]}]

set_clock_groups -asynchronous -group {adc_bclk_4} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_4} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_4};


# ADC #5
set_property SLEW FAST [get_ports {ADC_CNV_n[5]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[5]}]
set_property PACKAGE_PIN P20 [get_ports {ADC_CNV_n[5]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[5]}]
# External Termination
set_property PACKAGE_PIN R25 [get_ports {ADC_SCK_P[5]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[5]}]
# External Termination
set_property PACKAGE_PIN P25 [get_ports {ADC_SCK_N[5]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[5]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[5]}]
set_property PACKAGE_PIN N21 [get_ports {ADC_SCKOUT_P[5]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[5]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[5]}]
set_property PACKAGE_PIN N22 [get_ports {ADC_SCKOUT_N[5]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[5]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[5]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[5]}]
set_property PACKAGE_PIN N26 [get_ports {ADC_SDO1_P[5]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[5]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[5]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[5]}]
set_property PACKAGE_PIN M26 [get_ports {ADC_SDO1_N[5]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[5]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[5]}]
```

```
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[5]}]
set_property PACKAGE_PIN R26 [get_ports {ADC_SDO2_P[5]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[5]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[5]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[5]}]
set_property PACKAGE_PIN P26 [get_ports {ADC_SDO2_N[5]}]


# Timings
create_clock -name adc_bclk_5 -period 9.523809523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[5]}]

set_input_delay -clock adc_bclk_5 -min 0 [get_ports {ADC_SDO1_P[5]}]
set_input_delay -clock adc_bclk_5 -max 2 [get_ports {ADC_SDO1_P[5]}]
set_input_delay -clock adc_bclk_5 -min 0 [get_ports {ADC_SDO2_P[5]}]
set_input_delay -clock adc_bclk_5 -max 2 [get_ports {ADC_SDO2_P[5]}]

set_clock_groups -asynchronous -group {adc_bclk_5} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_5} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_5};


# ADC #6
set_property SLEW FAST [get_ports {ADC_CNV_n[6]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[6]}]
set_property PACKAGE_PIN Y25 [get_ports {ADC_CNV_n[6]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[6]}]
# External Termination
set_property PACKAGE_PIN W23 [get_ports {ADC_SCK_P[6]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[6]}]
# External Termination
set_property PACKAGE_PIN W24 [get_ports {ADC_SCK_N[6]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[6]}]
set_property PACKAGE_PIN AA23 [get_ports {ADC_SCKOUT_P[6]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[6]}]
set_property PACKAGE_PIN AB24 [get_ports {ADC_SCKOUT_N[6]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[6]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[6]}]
set_property PACKAGE_PIN V23 [get_ports {ADC_SDO1_P[6]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[6]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[6]}]
set_property PACKAGE_PIN V24 [get_ports {ADC_SDO1_N[6]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[6]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[6]}]
set_property PACKAGE_PIN U22 [get_ports {ADC_SDO2_P[6]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[6]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[6]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[6]}]
set_property PACKAGE_PIN V22 [get_ports {ADC_SDO2_N[6]}]


# Timings
create_clock -name adc_bclk_6 -period 9.523809523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[6]}]

set_input_delay -clock adc_bclk_6 -min 0 [get_ports {ADC_SDO1_P[6]}]
set_input_delay -clock adc_bclk_6 -max 2 [get_ports {ADC_SDO1_P[6]}]
set_input_delay -clock adc_bclk_6 -min 0 [get_ports {ADC_SDO2_P[6]}]
set_input_delay -clock adc_bclk_6 -max 2 [get_ports {ADC_SDO2_P[6]}]

set_clock_groups -asynchronous -group {adc_bclk_6} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_6} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_6};


# ADC #7
set_property SLEW FAST [get_ports {ADC_CNV_n[7]}]
```

```
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[7]}]
set_property PACKAGE_PIN M22 [get_ports {ADC_CNV_n[7]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[7]}]
# External Termination
set_property PACKAGE_PIN U19 [get_ports {ADC_SCK_P[7]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[7]}]
# External Termination
set_property PACKAGE_PIN U20 [get_ports {ADC_SCK_N[7]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[7]}]
set_property PACKAGE_PIN R22 [get_ports {ADC_SCKOUT_P[7]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[7]}]
set_property PACKAGE_PIN R23 [get_ports {ADC_SCKOUT_N[7]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[7]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[7]}]
set_property PACKAGE_PIN AD26 [get_ports {ADC_SDO1_P[7]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[7]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[7]}]
set_property PACKAGE_PIN AE26 [get_ports {ADC_SDO1_N[7]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[7]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[7]}]
set_property PACKAGE_PIN P24 [get_ports {ADC_SDO2_P[7]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[7]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[7]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[7]}]
set_property PACKAGE_PIN N24 [get_ports {ADC_SDO2_N[7]}]


# Timings
create_clock -name adc_bclk_7 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[7]}]

set_input_delay -clock adc_bclk_7 -min 0 [get_ports {ADC_SDO1_P[7]}]
set_input_delay -clock adc_bclk_7 -max 2 [get_ports {ADC_SDO1_P[7]}]
set_input_delay -clock adc_bclk_7 -min 0 [get_ports {ADC_SDO2_P[7]}]
set_input_delay -clock adc_bclk_7 -max 2 [get_ports {ADC_SDO2_P[7]}]

set_clock_groups -asynchronous -group {adc_bclk_7} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_7} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_7};


# ADC #8
set_property SLEW FAST [get_ports {ADC_CNV_n[8]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[8]}]
set_property PACKAGE_PIN AC26 [get_ports {ADC_CNV_n[8]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[8]}]
# External Termination
set_property PACKAGE_PIN AE23 [get_ports {ADC_SCK_P[8]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[8]}]
# External Termination
set_property PACKAGE_PIN AF23 [get_ports {ADC_SCK_N[8]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[8]}]
set_property PACKAGE_PIN Y22 [get_ports {ADC_SCKOUT_P[8]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[8]}]
set_property PACKAGE_PIN AA22 [get_ports {ADC_SCKOUT_N[8]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[8]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[8]}]
set_property PACKAGE_PIN AB22 [get_ports {ADC_SDO1_P[8]}]
```

```
set_property SLEW FAST [get_ports {ADC_SDO1_N[8]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[8]}]
set_property PACKAGE_PIN AC22 [get_ports {ADC_SDO1_N[8]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[8]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[8]}]
set_property PACKAGE_PIN AD23 [get_ports {ADC_SDO2_P[8]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[8]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[8]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[8]}]
set_property PACKAGE_PIN AD24 [get_ports {ADC_SDO2_N[8]}]


# Timings
create_clock -name adc_bclk_8 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[8]}]

set_input_delay -clock adc_bclk_8 -min 0 [get_ports {ADC_SDO1_P[8]}]
set_input_delay -clock adc_bclk_8 -max 2 [get_ports {ADC_SDO1_P[8]}]
set_input_delay -clock adc_bclk_8 -min 0 [get_ports {ADC_SDO2_P[8]}]
set_input_delay -clock adc_bclk_8 -max 2 [get_ports {ADC_SDO2_P[8]}]

set_clock_groups -asynchronous -group {adc_bclk_8} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_8} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_8};


# ADC #9
set_property SLEW FAST [get_ports {ADC_CNV_n[9]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[9]}]
set_property PACKAGE_PIN U16 [get_ports {ADC_CNV_n[9]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[9]}]
# External Termination
set_property PACKAGE_PIN AA25 [get_ports {ADC_SCK_P[9]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[9]}]
# External Termination
set_property PACKAGE_PIN AB25 [get_ports {ADC_SCK_N[9]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[9]}]
set_property PACKAGE_PIN G17 [get_ports {ADC_SCKOUT_P[9]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[9]}]
set_property PACKAGE_PIN F18 [get_ports {ADC_SCKOUT_N[9]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[9]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[9]}]
set_property PACKAGE_PIN F19 [get_ports {ADC_SDO1_P[9]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[9]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[9]}]
set_property PACKAGE_PIN E20 [get_ports {ADC_SDO1_N[9]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[9]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[9]}]
set_property PACKAGE_PIN H19 [get_ports {ADC_SDO2_P[9]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[9]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[9]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[9]}]
set_property PACKAGE_PIN G20 [get_ports {ADC_SDO2_N[9]}]


# Timings
create_clock -name adc_bclk_9 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[9]}]

set_input_delay -clock adc_bclk_9 -min 0 [get_ports {ADC_SDO1_P[9]}]
set_input_delay -clock adc_bclk_9 -max 2 [get_ports {ADC_SDO1_P[9]}]
set_input_delay -clock adc_bclk_9 -min 0 [get_ports {ADC_SDO2_P[9]}]
set_input_delay -clock adc_bclk_9 -max 2 [get_ports {ADC_SDO2_P[9]}]
```

```
set_clock_groups -asynchronous -group {adc_bclk_9} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_9} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_9};


# ADC #10
set_property SLEW FAST [get_ports {ADC_CNV_n[10]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[10]}]
set_property PACKAGE_PIN L24 [get_ports {ADC_CNV_n[10]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[10]}]
# External Termination
set_property PACKAGE_PIN W25 [get_ports {ADC_SCK_P[10]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[10]}]
# External Termination
set_property PACKAGE_PIN W26 [get_ports {ADC_SCK_N[10]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[10]}]
set_property PACKAGE_PIN E18 [get_ports {ADC_SCKOUT_P[10]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[10]}]
set_property PACKAGE_PIN D18 [get_ports {ADC_SCKOUT_N[10]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[10]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[10]}]
set_property PACKAGE_PIN A18 [get_ports {ADC_SDO1_P[10]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[10]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[10]}]
set_property PACKAGE_PIN A19 [get_ports {ADC_SDO1_N[10]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[10]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[10]}]
set_property PACKAGE_PIN C16 [get_ports {ADC_SDO2_P[10]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[10]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[10]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[10]}]
set_property PACKAGE_PIN B16 [get_ports {ADC_SDO2_N[10]}]


# Timings
create_clock -name adc_bclk_10 -period 9.523809523809523809523809523809523809 [get_ports {ADC_SCKOUT_P[10]}]

set_input_delay -clock adc_bclk_10 -min 0 [get_ports {ADC_SDO1_P[10]}]
set_input_delay -clock adc_bclk_10 -max 2 [get_ports {ADC_SDO1_P[10]}]
set_input_delay -clock adc_bclk_10 -min 0 [get_ports {ADC_SDO2_P[10]}]
set_input_delay -clock adc_bclk_10 -max 2 [get_ports {ADC_SDO2_P[10]}]

set_clock_groups -asynchronous -group {adc_bclk_10} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_10} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_10};


# ADC #11
set_property SLEW FAST [get_ports {ADC_CNV_n[11]}]
set_property IOSTANDARD LVCMOS25 [get_ports {ADC_CNV_n[11]}]
set_property PACKAGE_PIN M21 [get_ports {ADC_CNV_n[11]}]


set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_P[11]}]
# External Termination
set_property PACKAGE_PIN K25 [get_ports {ADC_SCK_P[11]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCK_N[11]}]
# External Termination
set_property PACKAGE_PIN K26 [get_ports {ADC_SCK_N[11]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_P[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_P[11]}]
set_property PACKAGE_PIN P23 [get_ports {ADC_SCKOUT_P[11]}]

set_property IOSTANDARD LVDS_25 [get_ports {ADC_SCKOUT_N[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SCKOUT_N[11]}]
```

```
set_property PACKAGE_PIN N23 [get_ports {ADC_SCKOUT_N[11]}]


set_property SLEW FAST [get_ports {ADC_SDO1_P[11]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_P[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_P[11]}]
set_property PACKAGE_PIN K20 [get_ports {ADC_SDO1_P[11]}]

set_property SLEW FAST [get_ports {ADC_SDO1_N[11]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO1_N[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO1_N[11]}]
set_property PACKAGE_PIN J20 [get_ports {ADC_SDO1_N[11]}]

set_property SLEW FAST [get_ports {ADC_SDO2_P[11]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_P[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_P[11]}]
set_property PACKAGE_PIN M25 [get_ports {ADC_SDO2_P[11]}]

set_property SLEW FAST [get_ports {ADC_SDO2_N[11]}]
set_property IOSTANDARD LVDS_25 [get_ports {ADC_SDO2_N[11]}]
set_property DIFF_TERM TRUE [get_ports {ADC_SDO2_N[11]}]
set_property PACKAGE_PIN L25 [get_ports {ADC_SDO2_N[11]}]


# Timings
create_clock -name adc_bclk_11 -period 9.5238095238095238095238095238095 [get_ports {ADC_SCKOUT_P[11]}]

set_input_delay -clock adc_bclk_11 -min 0 [get_ports {ADC_SDO1_P[11]}]
set_input_delay -clock adc_bclk_11 -max 2 [get_ports {ADC_SDO1_P[11]}]
set_input_delay -clock adc_bclk_11 -min 0 [get_ports {ADC_SDO2_P[11]}]
set_input_delay -clock adc_bclk_11 -max 2 [get_ports {ADC_SDO2_P[11]}]

set_clock_groups -asynchronous -group {adc_bclk_11} -group {USER_CLKA};

set_max_delay 10 -datapath_only -from {adc_bclk_11} -to {USER_CLKA};
set_max_delay 10 -datapath_only -from {USER_CLKA} -to {adc_bclk_11};


## ############################################################################################### ##
## Section: Front I/O
## ############################################################################################### ##

# Differential Trigger Inputs
set_property IOSTANDARD LVDS_25 [get_ports {DIFF_K7_P[0]}]
set_property DIFF_TERM TRUE [get_ports {DIFF_K7_P[0]}]
set_property PACKAGE_PIN L19 [get_ports {DIFF_K7_P[0]}]

set_property IOSTANDARD LVDS_25 [get_ports {DIFF_K7_N[0]}]
set_property DIFF_TERM TRUE [get_ports {DIFF_K7_N[0]}]
set_property PACKAGE_PIN L20 [get_ports {DIFF_K7_N[0]}]


set_property IOSTANDARD LVDS_25 [get_ports {DIFF_K7_P[1]}]
set_property DIFF_TERM TRUE [get_ports {DIFF_K7_P[1]}]
set_property PACKAGE_PIN J18 [get_ports {DIFF_K7_P[1]}]

set_property IOSTANDARD LVDS_25 [get_ports {DIFF_K7_N[1]}]
set_property DIFF_TERM TRUE [get_ports {DIFF_K7_N[1]}]
set_property PACKAGE_PIN J19 [get_ports {DIFF_K7_N[1]}]


set_property IOSTANDARD LVDS [get_ports {DIFF_K7_P[2]}]
# External Termination
set_property PACKAGE_PIN AA4 [get_ports {DIFF_K7_P[2]}]

set_property IOSTANDARD LVDS [get_ports {DIFF_K7_N[2]}]
# External Termination
set_property PACKAGE_PIN AB4 [get_ports {DIFF_K7_N[2]}]


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

# I2C Interface (inter-device communication)
set_property SLEW FAST [get_ports FPGA_SDA]
set_property IOSTANDARD LVCMOS33 [get_ports FPGA_SDA]
set_property PACKAGE_PIN G26 [get_ports FPGA_SDA]

set_property SLEW FAST [get_ports FPGA_SCL]
set_property IOSTANDARD LVCMOS33 [get_ports FPGA_SCL]
set_property PACKAGE_PIN F25 [get_ports FPGA_SCL]
```

```
# LEDs
set_property SLEW FAST [get_ports {USER_LED[0]}]
set_property IOSTANDARD LVCMOS33 [get_ports {USER_LED[0]}]
set_property PACKAGE_PIN J26 [get_ports {USER_LED[0]}]

set_property SLEW FAST [get_ports {USER_LED[1]}]
set_property IOSTANDARD LVCMOS33 [get_ports {USER_LED[1]}]
set_property PACKAGE_PIN E26 [get_ports {USER_LED[1]}]
```