The Embedded I/O Company

TEWS TECHNOLOGIES

TPMC634

Re-Configurable FPGA with

64 TTL I/O / 32 Differential I/O

Version 1.0

User Manual

Issue 1.0.1

April 2018

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

TPMC634-10R

Re-Configurable FPGA with 64 TTL I/O

TPMC634-11R

Re-Configurable FPGA with 32 Differential EIA-422 / EIA-485 I/O

TPMC634-12R

Re-Configurable FPGA with 16 Differential EIA-422 / EIA-485 I/O and 32 TTL I/O

TPMC634-13R

Re-Configurable FPGA with 32 Differential M-LVDS I/O

TPMC634-14R

Re-Configurable FPGA with 16 Differential M-LVDS I/O and 32 TTL I/O

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	April 2015
1.0.1	Order Option Description unified	April 2018

Table of Contents

List of Figures

List of Tables

1 Product Description

The TPMC634 is a standard single-width 32 bit PMC module providing a user programmable FPGA with front-I/O and P14 rear-I/O.

The TPMC634 provides a 32 bit 33MHz PCI Target interface. A dedicated (FPGA based) PCI Target Chip is used as a target bridge between the PCI bus and an on-board off-chip local bus. The local bus signals are available at User FPGA I/O pins, so the User FPGA logic design adapts to the local bus and is spared from implementing a PCI interface adaption.

The user programmable FPGA is a Xilinx Spartan-6 part number XC6SLX25-2-FGG484I.

The User FPGA is typically auto-configured by an on-board serial SPI Flash device after power-up. The serial SPI Flash device is in-system programmable via the PCI bus or via the JTAG header (indirectly via FPGA I/O pins using the Xilinx iMPACT software and platform cable programmer hardware).

Other options for volatile FPGA configuration are via JTAG (JTAG header) or in-system direct FPGA programming via the PCI bus.

An in-system debugging option is available via the on-board JTAG header. After power-up, the FPGA always attempts auto-configuration from the on-board SPI Flash.

There are five order options which are differing regarding the I/O interface:

- The TPMC634-10R provides 64 single-ended TTL I/O lines
- The TPMC634-11R provides 32 differential EIA-422 / EIA-485 I/O lines
- The TPMC634-12R provides a mix of 16 differential EIA-422 / EIA-485 I/O lines and 32 single-ended TTL I/O lines
- The TPMC634-13R provides 32 differential M-LVDS I/O lines
- The TPMC634-14R provides a mix of 16 differential M-LVDS I/O lines and 32 single-ended TTL I/O lines

The I/O line transmitters can be enabled or disabled per I/O line. The I/O line receivers are always enabled, so each I/O line level can always be monitored.

All I/O lines are ESD protected.

All I/O lines are available on both a 68 pin front-I/O connector and on the 64 pin P14 rear-I/O connector.

The TPMC634 I/O interface circuit and I/O connector pin assignment is compatible to the former TPMC630.

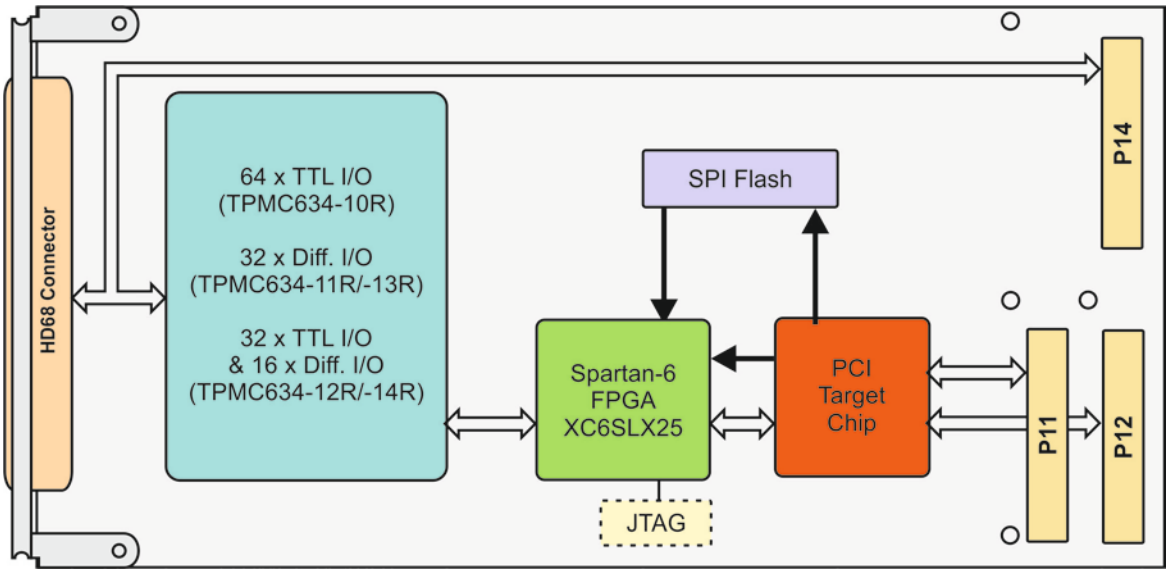


Figure 1-1 : TPMC634 Block Diagram

2 Technical Specification

PMC Interface		
Mechanical Interface	PCI Mezzanine Card (PMC) Interface confirming to IEEE P1386/P1386.1 Single Size	
Electrical Interface	PCI Rev. 3.0 compatible 33 MHz / 32 bit PCI Minimum PCI Clock Frequency is 8 MHz 3.3V and 5V PCI Signaling Voltage	
On-Board Devices		
PCI Target Chip (FPGA)	Lattice MachXO2 LCMXO2-4000HE-5BG320I	
User Programmable FPGA	Xilinx Spartan-6 XC6SLX25-2FGG484I	
EEPROM	M93C56WMN6 (or compatible)	
SPI Flash	Winbond W25Q32FVZPIG (W25Q32BV compatible)	
I/O Interface		
I/O Channels	TPMC634-10R: 64 TTL I/O TPMC634-11R: 32 RS422/485 Differential I/O TPMC634-12R: 32 TTL I/O and 16 RS422/485 Differential I/O TPMC634-13R: 32 M-LVDS Differential I/O TPMC634-14R: 32 TTL I/O and 16 M-LVDS Differential I/O	
I/O Transceiver	TPMC634-10R: 74LVT126 (or compatible) TPMC634-11R: MAX3078E (or compatible) TPMC634-12R: 74LVT126 & MAX3078E (or compatible) TPMC634-13R: SN65MLVD206 (or compatible) TPMC634-14R: 74LVT126 & SN65MLVD206 (or compatible)	
I/O Connector	Front I/O HD68 SCSI-3 Type Connector PMC P14 I/O (64 pin Mezzanine Connector)	
Physical Data		
Power Requirements	The +3.3V and ±12V from the PMC connector are not used. On-board I/O termination resistor load included, User FPGA example application running, without any external I/O load: TPMC634-10R: up to 180mA typical @ +5V DC TPMC634-11R: up to 620mA typical @ +5V DC TPMC634-12R: up to 410mA typical @ +5V DC TPMC634-13R: up to 300mA typical @ +5V DC (estimation) TPMC634-14R: up to 150mA typical @ +5V DC (estimation) Approximate values with external load on all I/O channels: TPMC634-10R: 1.0A @ +5V DC TPMC634-11R: 1.0A @ +5V DC TPMC634-12R: 1.0A @ +5V DC TPMC634-13R: 0.6A @ +5V DC (estimation) TPMC634-14R: 0.8A @ +5V DC (estimation)	
Temperature Range	Operating Storage	-40°C to +85°C -40°C to +85°C

MTBF	TPMC634-10R: 423000 h TPMC634-11R: 440000 h TPMC634-12R: 431000 h TPMC634-13R: 429000 h TPMC634-14R: 425000 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	84 g (all variants)

Table 2-1 : Technical Specification

3 Handling and Operation Instructions

3.1 ESD Protection

The PMC module is sensitive to static electricity. Packing, unpacking and all other module handling has to be done with the appropriate care.

3.2 User FPGA Power Dissipation Limit

The absolute maximum junction temperature of the Xilinx Spartan-6 XC6SLX25-2FGG484I is +125°C and must not be exceeded.

The recommended maximum junction temperature of the Xilinx Spartan-6 XC6SLX25-2FGG484I is +100°C and should not be exceeded.

The user must use appropriate design tools to ensure that the Spartan-6 FPGA junction temperature stays within the given limits for the actual User FPGA logic design and environment/system conditions (ambient temperature, air flow).

3.3 I/O Interface Installation

The TPMC634 I/O interface signals are available on both the 68 pin front-I/O connector and the 64 pin P14 rear-I/O connector.

Only one TPMC634 I/O interface must be used / installed at a time, either the front-I/O interface or the rear-I/O interface!

Do not use / install both the front I/O interface and the rear I/O interface at the same time!

3.4 Default TTL I/O Line States

The TPMC634 I/O interface transceivers are disabled by default.

However, the TTL I/O lines have on-board pull-to-reference resistors right at the PMC I/O interface. Hence, TTL I/O lines will show a valid logic level after power-up.

The common default reference level for all TTL I/O lines is configurable by the on-board rotary switch (3.3V, 5V or GND).

3.5 Pre-Installed User FPGA Example

The TPMC634 comes with a factory default User FPGA Example application stored in the on-board SPI Flash.

After power-up, the Spartan-6 User FPGA is automatically configured with the User FPGA Example application.

The User FPGA Example application provides registers for enabling the TPMC634 I/O line drivers!

Care must be taken, not to write accidently data to the registers implemented by the User FPGA Example application!

A brief description of the User FPGA Example register functions is appended to this user manual.

4 PCI Target Interface

4.1 PCI Configuration Space (PCI Header)

PCI CFG Reg. Offs.	Description				Config. by EEP	Pre-Configured Values
	31 24	23 16	15 8	7 0		
0x00	Device ID (TPMC634)		Vendor ID (TEWS Technologies)		Y	0x027A1498
0x04	Status		Command		N	--
0x08	Class Code			Revision ID	Y	0x11800001
0x0C	BIST not supported	Header Type	Latency Timer not supported	Cache Line Size not supported	N	0
0x10	Base Address Register 0 (BAR0) PCI Target Register Space (256 Byte)				N	0xFFFFFF00 (256 Byte)
0x14	Base Address Register 1 (BAR1) In-System Programming Space (SPI Flash) (256 Byte)				N	0xFFFFFF00 (256 Byte)
0x18	Base Address Register 2 (BAR2) ***[optional]*** User Space 0 (max 16 Mbyte) Programmable via Configuration EEPROM				Y	0xFFFC0000 (256 Kbyte)
0x1C	Base Address Register 3 (BAR3) ***[optional]*** User Space 1 (max 16 Mbyte) Programmable via Configuration EEPROM				Y	0x00000000
0x20	Base Address Register 4 (BAR4) ***[optional]*** User Space 2 (max 16 Mbyte) Programmable via Configuration EEPROM				Y	0x00000000
0x24	Base Address Register 5 (BAR5) ***[optional]*** User Space 3 (max 16 Mbyte) Programmable via Configuration EEPROM				Y	0x00000000
0x28	PCI CardBus Information Structure Pointer not supported				N	0
0x2C	Subsystem ID See below		Subsystem Vendor ID (TEWS Technologies)		Y	s.b. | 0x1498
0x30	Base Address for Local Expansion ROM not supported				N	0
0x34	Reserved			New Cap. Ptr.	N	0
0x38	Reserved				N	-
0x3C	Max_Lat	Min_Gnt	Interrupt Pin	Interrupt Line	N	0x00000100
0x40 - 0xFF	Reserved				N	0

Subsystem-ID: TPMC634-10R = 0x000A, TPMC634-11R = 0x000B, TPMC634-12R = 0x000C
TPMC634-13R = 0x000D, TPMC634-14R = 0x000E

Figure 4-1 : PCI Header

4.2 PCI BAR Overview

PCI BAR	PCI Space	Size (Byte)	Prefetch-able	Port Width (Bit)	Endian Mode	Description
0	MEM	256	No	32	Little	PCI Target Register Space
1	MEM	256	No	32	Little	In-System Programming Space (SPI Flash)
2	MEM	256K by default, 16M max	No	32	Little	Optional User Space 0 Enabled by factory default
3	MEM	Disabled by default, 16M max	No	32	Little	Optional User Space 1 Disabled by factory default
4	MEM	Disabled by default, 16M max	No	32	Little	Optional User Space 2 Disabled by factory default
5	MEM	Disabled by default, 16M max	No	32	Little	Optional User Space 3 Disabled by factory default

Table 4-1 : TPMC634 PCI BAR Overview

PCI BAR 0 & 1 are always enabled and are not configurable by the user.

PCI BARs 2 … 5 are optional user spaces for accessing the User FPGA on the local bus. PCI BARs 2 … 5 are configurable (enable, size) via the Configuration EEPROM.

From the optional user spaces, only PCI BAR2 (user space 0) is enabled with a 256 KB size per factory default. This space implements the registers of the user FPGA example application which comes with the TPMC634 (a brief description of the user FPGA example is appended to this user manual). PCI BARs 3-5 are disabled per factory default.

Note that all TPMC634 PCI BAR spaces are operating in little endian mode only.

For PCI BAR 1 PCI read and write commands are terminated directly (within 16 PCI clock cycles) with Disconnect-with-Data.

For PCI BARs 0 & 2-5 a FIFO (up to 16 PCI commands) is used to pass read/write parameters to/from the PCI bus. PCI write commands are handled as single-cycle posted writes. PCI read commands are handled as single-cycle delayed reads. Incoming writes are accepted and queued to the FIFO. Delayed reads are handled one at a time queued to the FIFO. PCI commands are retried when the FIFO is full.

4.3 PCI Configuration EEPROM Parameter

The following PCI configuration space parameters are loaded from an on-board serial EEPROM after PCI reset.

Serial EEPROM Word Offset	Parameter	Default Value (Default EEPROM)	Fallback Value (Invalid EEPROM)
PCI Configuration Space Parameter			
0x00	Device ID	0x027A	0x027A
0x01	Vendor ID	0x1498	0x1498
0x02	Class Code	0x1180	0x1180
0x03	Class Code / PCI Rev	0x0001	0x0001
0x04	Subsystem ID	Card Variant	0x0000
0x05	Subsystem Vendor ID	0x1498	0x0000
0x06	Reserved	-	-
0x07	Reserved	-	-
0x08	MSW PCI BAR 2 / User Space 0	0xFFFC	0x0000
0x09	LSW PCI BAR 2 / User Space 0	0x0000	0x0000
0x0A	MSW PCI BAR 3 / User Space 1	0x0000	0x0000
0x0B	LSW PCI BAR 3 / User Space 1	0x0000	0x0000
0x0C	MSW PCI BAR 4 / User Space 2	0x0000	0x0000
0x0D	LSW PCI BAR 4 / User Space 2	0x0000	0x0000
0x0E	MSW PCI BAR 5 / User Space 3	0x0000	0x0000
0x0F	LSW PCI BAR 5 / User Space 3	0x0000	0x0000

Table 4-2 : PCI Configuration EEPROM Map

If any of the first two EEPROM words is read as 0xFFFF the EEPROM data is discarded and the listed parameters are set to their fallback values. If both first EEPROM words are differing from 0xFFFF the listed parameters are set according to the EEPROM content.

Note that for an invalid/blank EEPROM, the card is still identified as a TPMC634, but there is no valid card variant coding and there also is no PCI BAR enabled for the optional user space.

If the User FPGA logic design only requires a single PCI BAR space and not more than 256 Kbyte address space, there usually is no need to alter the factory default Configuration EEPROM content.

See the Configuration EEPROM Register description for details about how to read from or write to the Configuration EEPROM.

4.4 PCI Clock Frequency

The TPMC634 provides a 32 bit 33 MHz PCI Target Interface. The minimum PCI clock frequency is 8 MHz (lower PCI clock frequencies are not supported).

4.5 PCI Access Times

The following table shows the approximate TPMC634 PCI access times. Two PCI bus idle cycles after each PCI access (and between delayed read retries) and 0 address wait states, 1 data wait state on the local bus are assumed.

PCI BAR	BAR Description	Write [PCI Clock Cycles] (Time @ 33MHz)	Read [PCI Clock Cycles] (Time @ 33MHz)
BAR0	PCI Target Register Space	5 + 2 Idle (212ns)	14 + 2 Idle (485ns)
BAR1	In-System Programming Space	5 + 2 Idle (212ns)	9 + 2 Idle (333ns)
BAR2-5	User Space 0-3	Posted Write on PCI Bus: 5 + 2 Idle (212ns) Additional Time for the write to complete on the Local Bus is approx. 120ns.	Delayed Read on PCI Bus: 22 + 2 Idle (727ns)

Table 4-3 : Approximate PCI Access Times

5 Address Maps

5.1 PCI Target Register Space

PCI BAR 0 implements the Configuration Registers located within the PCI Target Device.

Offset to PCI BAR	Register Name
0x90	Local Bus Interface Control / Status Register
0x94	Reserved
0x98	Reserved
0x9C	Reserved
0xA0	Reserved (I/O Interface Control / Status Register)
0xA4	Reserved
0xA8	Reserved
0xAC	Reserved
0xB0	Configuration EEPROM Register
0xB4	Reserved
0xB8	Reserved
0xBC	Reserved
0xC0	Interrupt Enable Register
0xC4	Interrupt Status Register
0xC8	Interrupt Configuration Register
0xCC	Reserved
0xD0	User FPGA Configuration Control & Status Register
0xD4	User FPGA Configuration Data Register (Slave SelectMAP)
0xD8	Reserved
0xDC	Reserved
0xE0	ISP Control Register (SPI)
0xE4	ISP Configuration Register (SPI)
0xE8	ISP Command Register (SPI)
0xEC	ISP Status Register (SPI)
0xF0	Control & Status Register
0xF4	Reserved
0xF8	Reserved
0xFC	Firmware Version Register

Table 5-1 : PCI Target Register Space

5.1.1 Local Bus Interface Register (0x90)

Bit	Symbol	Description	Access	Reset Value
Event Flags				
31:27	-	Reserved	-	0
26	LB_PLL_LOS	Local Bus PLL Loss-of-Lock Flag Set by HW when the local bus clock PLL lock status has changed from locked to not-locked Only functional for PLL based local bus clock implementation options Write '1' to clear the flag	R/C	0
25	LB_TERR	Local Bus Target Error Flag Set by HW in case the target reports an error condition during a local bus cycle Write '1' to clear the flag	R/C	0
24	LB_MABT	Local Bus Master Abort Flag (Time-Out) Set by HW in case of a master initiated local bus cycle abort (time-out) Only functional when the Local Bus Time-Out function is enabled Write '1' to clear the flag	R/C	0
Status Bits				
23:18	-	Reserved	-	0
17	LB_PLL_ST	Local Bus Clock PLL Lock Current Status 0: PLL not locked 1: PLL locked Only for PLL based local bus clock implementation options	R	x
16	LB_RST_ST	Local Bus Reset Line Status 0: Local Bus Reset Line is not active 1: Local Bus Reset Line is active Note that the local bus reset line is also active while the user FPGA is not configured or the local bus clock PLL is not locked	R	x
Control Bits				
15:3	-	Reserved	-	0
2	LB_PLL_RST	Local Bus Clock PLL Reset 0: PLL Operating Mode 1: PLL Reset Mode Only for PLL based local bus clock implementation options	R/W	0
1	LB_TO_DIS	Local Bus Time-Out Disable 0: Local Bus Time-Out Enabled (Default) 1: Local Bus Time-Out Disabled See the Local Bus Interface chapter for more information	R/W	0

0	LB_RST	Local Bus Reset 0: No SW controlled local bus reset 1: SW controlled local bus reset	R/W	0

Table 5-2 : Local Bus Interface Register

5.1.1 Configuration EEPROM Register (0xB0)

Bit	Symbol	Description	Access	Reset Value
31	EEP_GO_BSY	Configuration EEPROM Instruction Go / Busy Status Setting this bit starts performing the configured EEPROM instruction. The bit is cleared by hardware when the EEPROM instruction is done. While set, writes to this bit are ignored.	R/W (*)	0
30	-	Reserved	-	0
29:28	EEP_OP	Configuration EEPROM Instruction-Code 10 → READ 01 → WRITE 11 → ERASE 00 → WEN (EEP_A[7:6] = 11) WDS (EEP_A[7:6] = 00) ERAL (EEP_A[7:6] = 10) WRAL (EEP_A[7:6] = 01) WEN = Write Enable, WDS = Write Disable, ERAL = Erase All, WRAL = Write All	R/W	0
27:24	-	Reserved	-	0
23:16	EEP_A	Configuration EEPROM Address Address A[7:0] for x16 organized EEPROM Memory. 0x00 for first 16-bit EEPROM word, 0x01 for second 16-bit EEPROM word, etc. Required for READ, WRITE and ERASE Operation. Max address is 0x7F (2kbit EEPROM = 256 Byte = 128 EEPROM words). Address A[7:6] must be set accordingly for WEN | WDS | ERAL | WRAL EEPROM instructions	R/W	0
15:0	EEP_D	Configuration EEPROM Read/Write Data	R/W	0

Table 5-3 : Configuration EEPROM Register

Note that prior to a Write instruction an Erase instruction is not required.
Note that prior to a Write or Erase instruction a Write Enable instruction is required.

5.1.2 Interrupt Enable Register (0xC0)

Bit	Symbol	Description	Access	Reset Value
31:8	-	Reserved	-	0
Local Bus Interrupts				
7:6	-	Reserved	-	0
5	LB_ERR_IE	Local Bus Error Interrupt Enable 0: Interrupt Disabled 1: Interrupt Enabled	R/W	0
4	LB_USR_IE	Local Bus User Interrupt Enable 0: Interrupt Disabled 1: Interrupt Enabled	R/W	0
ISP Interrupts				
3:2	-	Reserved	-	0
1	ISP_INS_IE	ISP SPI Instruction Done Event Interrupt Enable 0: Interrupt Disabled 1: Interrupt Enabled	R/W	0
0	ISP_DAT_IE	ISP SPI Page Data Done Event Interrupt Enable 0: Interrupt Disabled 1: Interrupt Enabled	R/W	0

Table 5-4 : Interrupt Enable Register

Note that disabling an interrupt automatically clears any corresponding pending interrupt status.

5.1.1 Interrupt Status Register (0xC4)

Bit	Symbol	Description	Access	Reset Value
31:8	-	Reserved	-	0
Local Bus Interrupts				
7:6	-	Reserved	-	0
5	LB_ERR_IS	Local Bus Error Interrupt Status 0: Interrupt Status = Clear 1: Interrupt Status = Set If enabled, the interrupt status is set when any of the following error events occurs: • Local Bus PLL Loss-of-Lock Event • Local Bus Target Error Event • Local Bus Master Abort Event The Local Bus Interface Register may be read to determine which event(s) caused the interrupt. See the Interrupt Configuration Register for interrupt clearing.	R	0
4	LB_USR_IS	Local Bus User Interrupt Status If enabled, the interrupt is active while the User FPGA drives the LB_INT# signal low. 0: Interrupt Status = Clear 1: Interrupt Status = Set See the Interrupt Configuration Register for interrupt clearing.	R	0
ISP Interrupts				
3:2	-	Reserved	-	0
1	ISP_INS_IS	ISP SPI Instruction Done Event Interrupt Status 0: Interrupt Status = Clear 1: Interrupt Status = Set See the Interrupt Configuration Register for interrupt clearing.	R/C	0
0	ISP_DAT_IS	ISP SPI Page Data Done Event Interrupt Status 0: Interrupt Status = Clear 1: Interrupt Status = Set See the Interrupt Configuration Register for interrupt clearing.	R/C	0

Table 5-5 : Interrupt Status Register

When any Interrupt Status Register bit is set, the PCI INTA# interrupt is asserted.
Disabling an interrupt automatically clears the corresponding interrupt status.

5.1.2 Interrupt Configuration Register (0xC8)

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved	-	0
0	INT_ACK_MOD	Interrupt Acknowledge Mode 0: Interrupt Clear-by-Write Mode 1: Interrupt Clear-on-Read Mode **Interrupt Clear-by-Write Mode:** For both ISP based interrupts and for the Local Bus Error Interrupt, the interrupt status is cleared by writing a '1' to the corresponding Interrupt Status Register bit. The Local Bus User Interrupt status is cleared when the User FPGA stops driving the LB_INT# line low (or by disabling the interrupt). **Interrupt Clear-on-Read Mode:** When the Interrupt Status Register is read, any active ISP based interrupt status and Local Bus Error interrupt status is automatically cleared while the Local Bus User Interrupt is automatically disabled.	R/W	0

Table 5-6 : Interrupt Configuration Register

5.1.3 User FPGA Configuration Control/Status Register (0xD0)

Bit	Symbol	Description	Access	Reset Value
31:4	-	Reserved	-	0
3	FP_INIT_ STAT	User FPGA INIT_B Signal Status 0: FPGA INIT_B signal level is Low 1: FPGA INIT_B signal level is High Typically the FPGA INIT_B signal level is high after successful FPGA configuration.	R	x
2	FP_DONE_ STAT	User FPGA DONE Signal Status 0: FPGA DONE signal level is Low 1: FPGA DONE signal level is High The FPGA Done signal level is high in case of successful FPGA configuration.	R	x
1	FP_RE_CFG	User FPGA Re-Configuration Control **1: Prepare a User FPGA Re-Configuration:** Effects: The User FPGA PROG_B signal is asserted by HW. After a short time, the User FPGA asserts the INIT_B and DONE pins low and sets the general I/O pins to High-Z. The User FPGA Configuration Mode is set as configured by SW. After setting this bit, the SW should check the INIT_B signal status until it is read as '0'. **1 → 0: Start User FPGA Re-Configuration** Effects: After a short time the User FPGA releases the INIT_B pin, samples the configuration mode and starts the configuration sequence.	R/W	0

		After clearing this bit, the SW should check the INIT_B signal status until it is read as '1'. After power-up the FPGA automatically attempts to configure from the on-board SPI Flash in 'Master Serial / SPI' mode		
0	FP_CFG_MD	User FPGA Configuration Mode Control 0: Master Serial / SPI (configuration from SPI Flash) 1: Slave SelectMAP x8 (volatile configuration via PCI bus / software) This bit is processed while the User FPGA Re-Configuration Control Bit is set. The actual FPGA Re-Configuration is initiated by the User FPGA Re-Configuration Control Bit.	R/W	0

Table 5-7 : User FPGA Configuration Control/Status Register

Note that for User FPGA Re-Configuration from the SPI Flash, the ISP mode enable bit in the ISP Control Register must be clear.

5.1.4 User FPGA Configuration Data Register (Slave SelectMAP) (0xD4)

Bit	Symbol	Description	Access	Reset Value
31:0	FP_DAT	Write Data Register for direct volatile User FPGA programming in x8 Slave SelectMAP mode (not for SPI Flash programming) Must be written with 32-bit User FPGA programming data until the User FPGA Done pin goes high Each register write loads four configuration data bytes to the Spartan-6 User FPGA The byte order send to the Spartan-6 User FPGA is: 1. FP_CFG_DAT[7:0] 2. FP_CFG_DAT[15:8] 3. FP_CFG_DAT[23:16] 4. FP_CFG_DAT[31:24] E.g. the Xilinx Spartan-6 Sync Word 0xAA995566 must be mapped to the register byte lanes as follows: 0xAA → FP_CFG_DAT[7:0] 0x99 → FP_CFG_DAT[15:8] 0x55 → FP_CFG_DAT[23:16] 0x66 → FP_CFG_DAT[31:24] As required by the Xilinx Spartan-6 FPGA, each byte is bit-swapped by HW when sent to the Spartan-6 User FPGA After writing the actual programming data, writing dummy data (0xFFFFFFFF) may be required to generate additional configuration clock cycles	R/W	0

Table 5-8 : User FPGA Configuration Data Register (Slave SelectMAP)

5.1.1 ISP Control Register (SPI) (0xE0)

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved	-	0
0	ISP_EN	ISP Mode Enable 0: Disable ISP Mode 1: Enable ISP Mode This bit controls on-board analog multiplexers for signal connections between the MachXO2 FPGA, the User FPGA configuration interface and the on-board SPI Flash. Must be set to 1 for SPI Flash programming via PCI Must be set to 0 when the User FPGA should configure from the SPI Flash in 'Master Serial / SPI' mode (e.g. after SPI Flash programming)	R/W	0

Table 5-9 : ISP Control Register (SPI)

5.1.2 ISP Configuration Register (SPI) (0xE4)

Bit	Symbol	Description	Access	Reset Value
31:24	ISP_SPI_ADD	SPI Flash Address A7-A0	R/W	0
23:16		SPI Flash Address A15-A8	R/W	0
15:8		SPI Flash Address A23-A16	R/W	0
7:0	ISP_SPI_INS	SPI Flash Instruction Code Supported Instructions: 0x02 – SPI Flash Page Program 0x20 – SPI Flash Sector Erase 0x60 – SPI Flash Chip Erase 0x03 – SPI Flash Page Read 0x31 – SPI Flash Set QE Bit	R/W	0

Table 5-10: ISP Configuration Register (SPI)

5.1.3 ISP Command Register (SPI) (0xE8)

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved	-	0
1	ISP_SPI_ BRK_CMD	ISP SPI Break Instruction Command Bit Writing a '1' logs a request to break/cancel the current SPI Instruction. Check the Instruction Busy Bit in the ISP Status Register for SPI Instruction busy/done status. Ignored (lost) while the Instruction Busy Bit is clear in the ISP Status Register. Always read as '0'.	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'.	W	0

Table 5-11: ISP Command Register (SPI)

5.1.4 ISP Status Register (SPI) (0xEC)

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved	-	0
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 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 In-System Programming Space (in program mode) or SPI Flash page data is available in the In-System Programming 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 5-12: ISP Status Register (SPI)

5.1.1 Control & Status Register (0xF0)

Bit	Symbol	Description	Access	Reset Value
31:24	-	Reserved	R	0
23:17	-	Reserved Control Write as '0'	R/W	0
16	MXO_LED	MXO LED Control 0: MXO LED Off 1: MXO LED On (Default)	R/W	1
15:4	-	Reserved Status	R	0
3:0	DIP_SW	DIP-Switch Setting The bits are reflecting the on-board DIP switch configuration (ON = 1, OFF = 0) DIP_SW[0] is mapped to switch pos. 1. DIP_SW[1] is mapped to switch pos. 2. DIP_SW[2] is mapped to switch pos. 3. DIP_SW[3] is mapped to switch pos. 4.	R	-

Table 5-13: DIP-Switch Register

5.1.2 Firmware Version Register (0xFC)

Bit	Symbol	Description	Access	Reset Value
31:0	FW_VER	Version of the MachXO2 Firmware	R	-

Table 5-14: Version Register

5.2 In-System Programming Space

PCI BAR 1 is reserved as In-System Programming Space for in-system programming of the User FPGA Configuration Memory SPI Flash.

The In-System Programming Space size is 256 byte, covering an SPI Flash Memory Page. Supported SPI Flash read and write/program instructions are page-based.

For ISP write/program instructions, the data must be written (zero-based) to the In-System Programming 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 In-System Programming Space after the instruction is done. The data is passed for a complete SPI Flash memory page.

Control and status register for in-system SPI Flash programming are located in the PCI Target Register Space.

Note that passing programming data for direct volatile FPGA in-system programming in Slave SelectMAP configuration mode is not handled by the In-System Programming Space but by a dedicated data register in the PCI Target Register Space.

5.3 User Space(s)

PCI BARs 2 to 5 are reserved for implementing optional user spaces.

The TPMC634 supports up to four user spaces for accessing the user programmable Spartan-6 FPGA on the TPMC634 local bus. If enabled, user spaces must be used in ascending order.

Each user space is assigned to a dedicated PCI BAR. Each user space has a max size of 16MB and a dedicated select signal on the local bus interface. If required, the user FPGA logic design may use upper address lines for further address decoding within a local space.

User spaces are configured (enable & size) via the Configuration EEPROM. The Configuration EEPROM Register in the PCI Target Register Space is used for programming/reading the Configuration EEPROM.

By factory default, only user space 0 (PCI BAR 2) is enabled with a 256 Kbyte size.

The actual user space implementation depends on the user FPGA logic design.

PCI BAR	User Space	Local Bus Select Signal	Address Space Size
2	0	LB_SEL#[0]	16Mbyte max
3	1	LB_SEL#[1]	16Mbyte max
4	2	LB_SEL#[2]	16Mbyte max
5	3	LB_SEL#[3]	16Mbyte max

Table 5-15: User Space Overview

6 User Programmable FPGA

6.1 FPGA Part

The User FPGA on the TPMC634 is a Xilinx Spartan-6 XC6SLX25-2-FGG484I.

The Xilinx Spartan-6 XC6SLX25 is supported by the Xilinx ISE WebPACK tool.

The following table shows some key User FPGA resources.

FPGA Resource	XC6SLX25
Logic Cells	24051
CLBs	1879
Slices	3758
FlipFlops	30064
LUTs (6 Inputs)	15032
DistRAM (kb)	229 (approx. 28 Kbyte)
BRAM (kb)	936 (approx. 104 Kbyte)

Table 6-1 : User FPGA Resources

6.2 I/O Bank Supply & Supported I/O Standards

On the TPMC634, all Spartan-6 VCCO I/O bank power supplies are connected to the 3.3V supply.

The VCCAUX power supply is also connected to a 3.3V supply.

Supported I/O Standards are: LVTTL, LVCMOS33.

6.3 User FPGA I/O Signal & Pin Description

The following User FPGA I/O signal groups are available for the user application:

- Local Bus Interface Signals
- I/O Interface Signals
- Other User Signals

6.3.1 Local Bus Interface Signals

Local Bus Interface Signals						
Signal Name	Pin	Direction	IO Standard	IO Bank	Drive [mA]	Slew Rate
LB_AD[0]	R11	I/O	LVCMOS33	2	8	SLOW
LB_AD[1]	T11	I/O	LVCMOS33	2	8	SLOW
LB_AD[2]	AA10	I/O	LVCMOS33	2	8	SLOW

Local Bus Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
LB_AD[3]	AB10	I/O	LVCMOS33	2	8	SLOW
LB_AD[4]	V11	I/O	LVCMOS33	2	8	SLOW
LB_AD[5]	W11	I/O	LVCMOS33	2	8	SLOW
LB_AD[6]	Y9	I/O	LVCMOS33	2	8	SLOW
LB_AD[7]	AB9	I/O	LVCMOS33	2	8	SLOW
LB_AD[8]	W10	I/O	LVCMOS33	2	8	SLOW
LB_AD[9]	Y10	I/O	LVCMOS33	2	8	SLOW
LB_AD[10]	AA8	I/O	LVCMOS33	2	8	SLOW
LB_AD[11]	AB8	I/O	LVCMOS33	2	8	SLOW
LB_AD[12]	W8	I/O	LVCMOS33	2	8	SLOW
LB_AD[13]	V7	I/O	LVCMOS33	2	8	SLOW
LB_AD[14]	W9	I/O	LVCMOS33	2	8	SLOW
LB_AD[15]	Y8	I/O	LVCMOS33	2	8	SLOW
LB_AD[16]	U9	I/O	LVCMOS33	2	8	SLOW
LB_AD[17]	V9	I/O	LVCMOS33	2	8	SLOW
LB_AD[18]	W6	I/O	LVCMOS33	2	8	SLOW
LB_AD[19]	Y6	I/O	LVCMOS33	2	8	SLOW
LB_AD[20]	Y5	I/O	LVCMOS33	2	8	SLOW
LB_AD[21]	AB5	I/O	LVCMOS33	2	8	SLOW
LB_AD[22]	AA4	I/O	LVCMOS33	2	8	SLOW
LB_AD[23]	AB4	I/O	LVCMOS33	2	8	SLOW
LB_AD[24]	Y3	I/O	LVCMOS33	2	8	SLOW
LB_AD[25]	AB3	I/O	LVCMOS33	2	8	SLOW
LB_AD[26]	R9	I/O	LVCMOS33	2	8	SLOW
LB_AD[27]	R8;	I/O	LVCMOS33	2	8	SLOW
LB_AD[28]	T7	I/O	LVCMOS33	2	8	SLOW
LB_AD[29]	R7	I/O	LVCMOS33	2	8	SLOW
LB_AD[30]	U6	I/O	LVCMOS33	2	8	SLOW
LB_AD[31]	V5	I/O	LVCMOS33	2	8	SLOW
LB_SEL#[0]	Y19	In	LVCMOS33	2	N/A	N/A
LB_SEL#[1]	AB19	In	LVCMOS33	2	N/A	N/A
LB_SEL#[2]	W18	In	LVCMOS33	2	N/A	N/A
LB_SEL#[3]	Y18	In	LVCMOS33	2	N/A	N/A

Local Bus Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
LB_BE#[0]	V15	In	LVCMOS33	2	N/A	N/A
LB_BE#[1]	AA18	In	LVCMOS33	2	N/A	N/A
LB_BE#[2]	AB18	In	LVCMOS33	2	N/A	N/A
LB_BE#[3]	Y17	In	LVCMOS33	2	N/A	N/A
LB_RST#	AB17	In	LVCMOS33	2	N/A	N/A
LB_CLK	Y13	In	LVCMOS33	2	N/A	N/A
LB_CYC#	AA21	In	LVCMOS33	2	N/A	N/A
LB_R/W#	AA14	In	LVCMOS33	2	N/A	N/A
LB_D/A#	AB14	In	LVCMOS33	2	N/A	N/A
LB_MRDY#	Y16	In	LVCMOS33	2	N/A	N/A
LB_MABT#	V13	In	LVCMOS33	2	N/A	N/A
LB_TRDY#	W15	Out	LVCMOS33	2	8	SLOW
LB_TERR#	W13	Out	LVCMOS33	2	8	SLOW
LB_INT#	AB21	Out	LVCMOS33	2	8	SLOW

Table 6-2 : Local Bus Interface Signals (FPGA Parameters)

See the Local Bus Interface chapter for a Local Bus Interface signal description.

6.3.2 I/O Interface Signals

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_IN[0]	Y2	In	LVCMOS33	3	N/A	N/A
FPGA_IN[1]	W1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[2]	U4	In	LVCMOS33	3	N/A	N/A
FPGA_IN[3]	M4	In	LVCMOS33	3	N/A	N/A
FPGA_IN[4]	U3	In	LVCMOS33	3	N/A	N/A
FPGA_IN[5]	T1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[6]	P2	In	LVCMOS33	3	N/A	N/A
FPGA_IN[7]	N1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[8]	L3	In	LVCMOS33	3	N/A	N/A
FPGA_IN[9]	K1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[10]	H4	In	LVCMOS33	3	N/A	N/A
FPGA_IN[11]	H1	In	LVCMOS33	3	N/A	N/A

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_IN[12]	H6	In	LVCMOS33	3	N/A	N/A
FPGA_IN[13]	F1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[14]	E3	In	LVCMOS33	3	N/A	N/A
FPGA_IN[15]	D1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[16]	G6	In	LVCMOS33	3	N/A	N/A
FPGA_IN[17]	H8	In	LVCMOS33	3	N/A	N/A
FPGA_IN[18]	A2	In	LVCMOS33	3	N/A	N/A
FPGA_IN[19]	J1	In	LVCMOS33	3	N/A	N/A
FPGA_IN[20]	K5	In	LVCMOS33	3	N/A	N/A
FPGA_IN[21]	J4	In	LVCMOS33	3	N/A	N/A
FPGA_IN[22]	C5	In	LVCMOS33	0	N/A	N/A
FPGA_IN[23]	C6	In	LVCMOS33	0	N/A	N/A
FPGA_IN[24]	C7	In	LVCMOS33	0	N/A	N/A
FPGA_IN[25]	A8	In	LVCMOS33	0	N/A	N/A
FPGA_IN[26]	C9	In	LVCMOS33	0	N/A	N/A
FPGA_IN[27]	D8	In	LVCMOS33	0	N/A	N/A
FPGA_IN[28]	C13	In	LVCMOS33	0	N/A	N/A
FPGA_IN[29]	D12	In	LVCMOS33	0	N/A	N/A
FPGA_IN[30]	F13	In	LVCMOS33	0	N/A	N/A
FPGA_IN[31]	G13	In	LVCMOS33	0	N/A	N/A
FPGA_IN[32]	F14	In	LVCMOS33	0	N/A	N/A
FPGA_IN[33]	C14	In	LVCMOS33	0	N/A	N/A
FPGA_IN[34]	C15	In	LVCMOS33	0	N/A	N/A
FPGA_IN[35]	C16	In	LVCMOS33	0	N/A	N/A
FPGA_IN[36]	C17	In	LVCMOS33	0	N/A	N/A
FPGA_IN[37]	A18	In	LVCMOS33	0	N/A	N/A
FPGA_IN[38]	B10	In	LVCMOS33	0	N/A	N/A
FPGA_IN[39]	A11	In	LVCMOS33	0	N/A	N/A
FPGA_IN[40]	B12	In	LVCMOS33	0	N/A	N/A
FPGA_IN[41]	C19	In	LVCMOS33	1	N/A	N/A
FPGA_IN[42]	B22	In	LVCMOS33	1	N/A	N/A
FPGA_IN[43]	D19	In	LVCMOS33	1	N/A	N/A
FPGA_IN[44]	F19	In	LVCMOS33	1	N/A	N/A

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_IN[45]	C20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[46]	F20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[47]	E20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[48]	K17	In	LVCMOS33	1	N/A	N/A
FPGA_IN[49]	H20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[50]	G20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[51]	K22	In	LVCMOS33	1	N/A	N/A
FPGA_IN[52]	M21	In	LVCMOS33	1	N/A	N/A
FPGA_IN[53]	N22	In	LVCMOS33	1	N/A	N/A
FPGA_IN[54]	R20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[55]	T22	In	LVCMOS33	1	N/A	N/A
FPGA_IN[56]	V21	In	LVCMOS33	1	N/A	N/A
FPGA_IN[57]	N19	In	LVCMOS33	1	N/A	N/A
FPGA_IN[58]	W20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[59]	K20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[60]	H22	In	LVCMOS33	1	N/A	N/A
FPGA_IN[61]	J20	In	LVCMOS33	1	N/A	N/A
FPGA_IN[62]	AA16	In	LVCMOS33	2	N/A	N/A
FPGA_IN[63]	Y14	In	LVCMOS33	2	N/A	N/A
FPGA_OUT[0]	Y1	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[1]	T4	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[2]	V3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[3]	V2	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[4]	U1	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[5]	R3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[6]	P1	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[7]	M2	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[8]	L1	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[9]	K6	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[10]	H3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[11]	G3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[12]	H5	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[13]	G4	Out	LVCMOS33	3	8	SLOW

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_OUT[14]	E1	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[15]	C3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[16]	F5	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[17]	B2	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[18]	B3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[19]	M3	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[20]	K4	Out	LVCMOS33	3	8	SLOW
FPGA_OUT[21]	A3	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[22]	A5	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[23]	B6	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[24]	A7	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[25]	D9	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[26]	A9	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[27]	D10	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[28]	A13	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[29]	H12	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[30]	D13	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[31]	E14	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[32]	H14	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[33]	B14	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[34]	A15	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[35]	B16	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[36]	A17	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[37]	E16	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[38]	A10	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[39]	D11	Out	LVCMOS33	0	8	SLOW
FPGA_OUT[40]	T19	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[41]	B20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[42]	A20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[43]	D20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[44]	D21	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[45]	C22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[46]	H19	Out	LVCMOS33	1	8	SLOW

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_OUT[47]	E22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[48]	F21	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[49]	T20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[50]	G22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[51]	L20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[52]	M22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[53]	P21	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[54]	R22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[55]	U20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[56]	V22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[57]	P19	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[58]	W22	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[59]	K19	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[60]	M20	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[61]	K18	Out	LVCMOS33	1	8	SLOW
FPGA_OUT[62]	AB16	Out	LVCMOS33	2	8	SLOW
FPGA_OUT[63]	Y15	Out	LVCMOS33	2	8	SLOW
FPGA_OE[0]	W3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[1]	T3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[2]	M5	Out	LVCMOS33	3	8	SLOW
FPGA_OE[3]	V1	Out	LVCMOS33	3	8	SLOW
FPGA_OE[4]	T2	Out	LVCMOS33	3	8	SLOW
FPGA_OE[5]	R1	Out	LVCMOS33	3	8	SLOW
FPGA_OE[6]	N3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[7]	M1	Out	LVCMOS33	3	8	SLOW
FPGA_OE[8]	K2	Out	LVCMOS33	3	8	SLOW
FPGA_OE[9]	J6	Out	LVCMOS33	3	8	SLOW
FPGA_OE[10]	H2	Out	LVCMOS33	3	8	SLOW
FPGA_OE[11]	G1	Out	LVCMOS33	3	8	SLOW
FPGA_OE[12]	F2	Out	LVCMOS33	3	8	SLOW
FPGA_OE[13]	F3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[14]	D2	Out	LVCMOS33	3	8	SLOW
FPGA_OE[15]	C1	Out	LVCMOS33	3	8	SLOW

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_OE[16]	J7	Out	LVCMOS33	3	8	SLOW
FPGA_OE[17]	B1	Out	LVCMOS33	3	8	SLOW
FPGA_OE[18]	J3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[19]	L4	Out	LVCMOS33	3	8	SLOW
FPGA_OE[20]	K3	Out	LVCMOS33	3	8	SLOW
FPGA_OE[21]	A4	Out	LVCMOS33	0	8	SLOW
FPGA_OE[22]	D6	Out	LVCMOS33	0	8	SLOW
FPGA_OE[23]	A6	Out	LVCMOS33	0	8	SLOW
FPGA_OE[24]	B8	Out	LVCMOS33	0	8	SLOW
FPGA_OE[25]	C8	Out	LVCMOS33	0	8	SLOW
FPGA_OE[26]	D7	Out	LVCMOS33	0	8	SLOW
FPGA_OE[27]	C10	Out	LVCMOS33	0	8	SLOW
FPGA_OE[28]	E12	Out	LVCMOS33	0	8	SLOW
FPGA_OE[29]	F12	Out	LVCMOS33	0	8	SLOW
FPGA_OE[30]	H13	Out	LVCMOS33	0	8	SLOW
FPGA_OE[31]	F15	Out	LVCMOS33	0	8	SLOW
FPGA_OE[32]	D14	Out	LVCMOS33	0	8	SLOW
FPGA_OE[33]	A14	Out	LVCMOS33	0	8	SLOW
FPGA_OE[34]	D15	Out	LVCMOS33	0	8	SLOW
FPGA_OE[35]	A16	Out	LVCMOS33	0	8	SLOW
FPGA_OE[36]	B18	Out	LVCMOS33	0	8	SLOW
FPGA_OE[37]	D17	Out	LVCMOS33	0	8	SLOW
FPGA_OE[38]	C11	Out	LVCMOS33	0	8	SLOW
FPGA_OE[39]	C12	Out	LVCMOS33	0	8	SLOW
FPGA_OE[40]	A12	Out	LVCMOS33	0	8	SLOW
FPGA_OE[41]	B21	Out	LVCMOS33	1	8	SLOW
FPGA_OE[42]	A21	Out	LVCMOS33	1	8	SLOW
FPGA_OE[43]	F18	Out	LVCMOS33	1	8	SLOW
FPGA_OE[44]	D22	Out	LVCMOS33	1	8	SLOW
FPGA_OE[45]	G19	Out	LVCMOS33	1	8	SLOW
FPGA_OE[46]	H18	Out	LVCMOS33	1	8	SLOW
FPGA_OE[47]	J17	Out	LVCMOS33	1	8	SLOW
FPGA_OE[48]	F22	Out	LVCMOS33	1	8	SLOW

I/O Interface Signals						
Signal Name	**Pin**	**Direction**	**IO Standard**	**IO Bank**	**Drive [mA]**	**Slew Rate**
FPGA_OE[49]	J19	Out	LVCMOS33	1	8	SLOW
FPGA_OE[50]	K21	Out	LVCMOS33	1	8	SLOW
FPGA_OE[51]	L22	Out	LVCMOS33	1	8	SLOW
FPGA_OE[52]	N20	Out	LVCMOS33	1	8	SLOW
FPGA_OE[53]	P22	Out	LVCMOS33	1	8	SLOW
FPGA_OE[54]	T21	Out	LVCMOS33	1	8	SLOW
FPGA_OE[55]	U22	Out	LVCMOS33	1	8	SLOW
FPGA_OE[56]	M19	Out	LVCMOS33	1	8	SLOW
FPGA_OE[57]	P20	Out	LVCMOS33	1	8	SLOW
FPGA_OE[58]	L17	Out	LVCMOS33	1	8	SLOW
FPGA_OE[59]	H21	Out	LVCMOS33	1	8	SLOW
FPGA_OE[60]	L19	Out	LVCMOS33	1	8	SLOW
FPGA_OE[61]	J22	Out	LVCMOS33	1	8	SLOW
FPGA_OE[62]	W14	Out	LVCMOS33	2	8	SLOW
FPGA_OE[63]	AB15	Out	LVCMOS33	2	8	SLOW

Table 6-3 : I/O Interface Signals (FPGA Parameters)

FPGA_OE[x]	PMC_IO[x]	FPGA_IN[x]
High-Z, 0	Differential I/O: Transmitter Disabled Single-Ended I/O: Transmitter Disabled I/O line level determined by on-board pull resistors / rotary switch	FPGA_IN[x] reflects PMC_IO[x]
1	FPGA_OUT[x] controls PMC_IO[x]	

Table 6-4 : FPGA I/O Interface Signal Description

The FPGA_IN[63:0] signals are reflecting the PMC_IO[63:0] line state.

The FPGA_OUT[63:0] signals are setting the PMC_IO[63:0] state when the corresponding on-board I/O transmitters are enabled.

The FPGA_OE[63:0] signals are enabling or disabling the on-board I/O transmitters.

All FPGA_OE[63:0] signals have an on-board pull-down resistor, disabling the I/O transmitters when the FPGA_OE pins are High-Z.

See the I/O Interface chapter for more information.

For the TPMC634-11R/-13R order options (differential I/O only) the FPGA_IN[63:32] signals are not used and not driven. The user FPGA logic design should implement FPGA internal pull-down or pull-up resistors on the FPGA_IN[63:32] lines for the TPMC634-11R/-13R order options to keep these signals from floating

For the TPMC634-12R/-14R order options (mixed TTL and differential I/O) the FPGA_IN[31:16] signals are not used and not driven. The user FPGA logic design should implement FPGA internal pull-down or pull-up resistors on the FPGA_IN[31:16] lines for the TPMC634-12R/-14R order options to keep these signals from floating.

It would also be convenient to enable an FPGA internal pulldown or pullup resistor for each FPGA_IN[63:0] signal in the users FPGA logic design, e.g. in the User Constraint File as shown below.

```
#
# I/O Interface
#
NET "IO_IN<0>" LOC = Y2 | PULLDOWN;
NET "IO_IN<1>" LOC = W1 | PULLDOWN;
NET "IO_IN<2>" LOC = U4 | PULLDOWN;
…
…
```

TPMC634 Variants	PMC I/O	FPGA I/O	Unused FPGA I/O
TPMC634-10R	Single-Ended I/O: IO[63:0]	Single-Ended I/O: FPGA_IN[63:0], FPGA_OUT[63:0], FPGA_OE[63:0]	N/A
TPMC634-11R TPMC634-13R	Differential I/O: IO[31:0]+/-	Differential I/O: FPGA_IN[31:0], FPGA_OUT[31:0], FPGA_OE[31:0]	FPGA_IN[63:32] are floating FPGA_OUT[63:32], FPGA_OE[63:32] have no effect
TPMC634-12R TPMC634-14R	Differential I/O: IO[15:0]+/- Single-Ended I/O: IO[63:32]	Differential I/O: FPGA_IN[15:0], FPGA_OUT[15:0], FPGA_OE[15:0] Single-Ended I/O: FPGA_IN[63:32], FPGA_OUT[63:32], FPGA_OE[63:32]	FPGA_IN[31:16] are floating FPGA_OUT[31:16], FPGA_OE[31:16] have no effect

Table 6-5 : FPGA I/O Interface Signal Usage for TPMC634 Variants

6.3.3 Other User Signals

Signal Name	Pin	Direction	IO Standard	IO Bank	Drive [mA]	Slew Rate
BAUD_CLK	AB11	In	LVCMOS33	2	N/A	N/A
AUX_CLK	AB13	In	LVCMOS33	2	N/A	N/A

Table 6-6 : Other User Signals

The BAUD_CLK signal is a 7.3728 MHz clock signal that may be used to implement standard serial communication baud rates for the I/O interface.

The AUX_CLK signal is a general purpose 33.25 MHz ±5% clock signal.

6.3.4 Reserved FPGA I/O Pins

All FPGA I/O pins listed as reserved must not be used by the user application!

Signal Name	Pin	Direction	IO Standard	IO Bank	Drive [mA]	Slew Rate
FPGA I/O Pins Reserved for Configuration						
CCLK	Y21	I/O	-	2	-	-
M0	AA22	I/O	-	2	-	-
M1	U15	I/O	-	2	-	-
MOSI_CSI_B_MISO0	AB20	I/O	-	2	-	-
D0_DIN_MISO_MISO1	AA20	I/O	-	2	-	-
D1_MISO2	U14	I/O	-	2	-	-
D2_MISO3	U13	I/O	-	2	-	-
D3	AA6	I/O	-	2	-	-
D4	AB6	I/O	-	2	-	-
D5	W4	I/O	-	2	-	-
D6	Y4	I/O	-	2	-	-
D7	Y7	I/O	-	2	-	-
INIT_B	T6	I/O	-	2	-	-
CSO_B	T5	I/O	-	2	-	-
RDWR_B	AB7	I/O	-	2	-	-
Other Reserved FPGA I/O Pins						
MXO_S6_0	W12	I/O	-	2	-	-
MXO_S6_1	Y11	I/O	-	2	-	-
MXO_S6_2	AB12	I/O	-	2	-	-

MXO_S6_3	AA12	I/O	-	2	-	-
LB_CLK_OSC	Y12	I/O	-	2	-	-
LB_REQ#	AA2	I/O	-	2	-	-
LB_GNT#	AB2	I/O	-	2	-	-

Table 6-7 : Reserved FPGA I/O Pins

6.4 User FPGA Power Dissipation Limit

The parts used on the TPMC634 are specified for industrial temperature range (-40°C… +85°C).

The user must use appropriate design tools to ensure that the Spartan-6 FPGA junction temperature stays within the given limits for the actual environment and system conditions (maximum ambient temperature, system air flow / cooling) and the actual FPGA logic design. The Xilinx ISE design software provides the XPower Analyzer tool for such purpose.

Key Parameters are:

- System (PMC Slot) Air Flow (Cooling)
- Actual Ambient Temperature Range
- Actual User FPGA Logic Design

The absolute maximum junction temperature of the Xilinx Spartan-6 XC6SLX25-2FGG484I is +125°C and must not be exceeded!

The recommended maximum junction temperature of the Xilinx Spartan-6 XC6SLX25-2FGG484I is +100°C and should not be exceeded.

The user must use appropriate design tools to ensure that with the actual FPGA logic design the Spartan-6 FPGA junction temperature stays within the given limits for the actual environment and system conditions.

7 User Programmable FPGA Configuration

7.1 User FPGA Configuration Options

The TPMC634 provides the following options for configuring the User FPGA.

- FPGA Configuration from on-board SPI Flash (Master Serial / SPI FPGA Configuration Mode)

  The SPI Flash must be programmed accordingly before FPGA configuration. Either via the PCI bus or with the Xilinx iMPACT software and programmer cable via the TPMC634 JTAG header.

- FPGA Configuration via PCI/Software (Slave SelectMAP FPGA Configuration Mode)

  Since the FPGA internal configuration memory is SRAM based, the FPGA configuration will get lost when power is turned off.

- FPGA Configuration via JTAG Header (FPGA Configuration per JTAG Port)

  Since the FPGA internal configuration memory is SRAM based, the FPGA configuration will get lost when power is turned off.

After power-up, the Spartan-6 User FPGA always attempts to load the configuration bitstream from the on-board SPI Flash in *Master Serial* / *SPI* configuration mode.

Upon delivery, the SPI Flash contains a TPMC634 Spartan-6 User FPGA Example Application (a brief description is appended to this user manual).

7.2 FPGA Configuration from SPI Flash

7.2.1 Auto-Configuration at Power-Up

After power-up, the User FPGA always attempts to load the configuration bitstream from the on-board serial SPI Flash in Master Serial / SPI FPGA Configuration Mode.

Upon delivery, the SPI Flash contains a TPMC634 Spartan-6 User FPGA Example Application (a brief description is appended to this user manual).

7.2.2 SW Controlled Re-Configuration

Besides a power-cycle, an FPGA re-configuration from the on-board SPI Flash can also be initiated by software.

Steps for User FPGA Re-Configuration from the SPI Flash in Master Serial / SPI Configuration Mode

1. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash.

2. Write 0x00000002 to the User FPGA Configuration Control/Status Register (0xD0)

   This will assert the Spartan-6 FPGA PROGRAM_B pin and will prepare the FPGA for re-configuration in Master Serial / SPI configuration mode.

3. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until both bit 2 & 3 are clear

   I.e. wait until the Spartan-6 FPGA asserts the INIT_B and DONE pins low.

4. Write 0x00000000 to the User FPGA Configuration Control/Status Register (0xD0)

   This will release the Spartan-6 FPGA PROGRAM_B pin and will start the FPGA re-configuration in Master Serial / SPI configuration mode.

5. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until bit 2 is set

   I.e. wait until the Spartan-6 FPGA asserts the DONE pin high, indicating successful FPGA re-configuration.

7.2.3 SPI Flash Preparation

For successful User FPGA configuration from SPI Flash in Master Serial / SPI configuration mode, the SPI Flash must be programmed accordingly.

See the *SPI Flash Programming* chapter for more information.

7.2.4 FPGA Configuration Time

Two parameters are determining the required time for the User FPGA to configure from the on-board SPI Flash: the SPI Flash Bus Width and the Configuration Clock Source & Rate.

SPI Flash Bus Width

The User FPGA configuration memory device is a 32Mbit Winbond W25Q serial SPI Flash with Quad mode support.

The TPMC634 supports x4 SPI mode for FPGA configuration from the SPI Flash (for SPI Flash programming only x1 SPI mode is supported).

Configuration Clock Source

In Master Serial / SPI configuration mode, the User FPGA CCLK (Configuration Clock) pin is an output, driven by the FPGA.

The TPMC634 supports using the Spartan-6 internal CCLK oscillator (duty cycle 40/60, frequency tolerance ±50%) as the configuration clock source. 12MHz is the highest supported Configuration Rate setting (resulting in a configuration clock frequency range from 6MHz to 18MHz).

SPI Flash bus width and Spartan-6 FPGA configuration clock source & rate are configured during the Xilinx ISE design flow. These are BitGen options which are configurable in the Generate Programming File / Configuration Options process properties. The following options are supported: ConfigRate = 12, ExtMasterCclk_en = No, SPI_buswidth = 4

The following table shows the estimated TPMC634 User FPGA configuration time when configuring from the on-board SPI Flash.

FPGA Device	Configuration Bits	Configuration Source	Configuration Rate	Estimated FPGA Configuration Time
XC6SLX25	6440432	Spartan-6 Internal Osc.	12MHz +/- 50%	90ms … 269ms (SPI x4)

Table 7-1 : Estimated FPGA Configuration Time

7.3 FPGA Configuration via PCI/Software

The TPMC634 supports direct (volatile) FPGA In-System-Programming via the PCI bus in Slave SelectMAP FPGA configuration mode (x8).

7.3.1 Configuration Data Files

Programming the TPMC634 User FPGA directly in Slave SelectMAP configuration mode requires certain result files from the Xilinx ISE design flow.

The .BIT file is a binary configuration data file containing header information that should not be programmed into the FPGA. The file is typically used with the Xilinx iMPACT software, e.g. for direct (volatile) FPGA device programming via a JTAG programmer cable or as a source for generating an appropriately formatted PROM/Flash file (.MCS file).

The .BIN file is a binary configuration data file without header information. The file is typically used for programming FPGA configuration memory devices (e.g. SPI Flash device) via a microprocessor or other means.

The .bit file is usually generated by default. The .bin file generation must be enabled by a BitGen option (or in the ISE design flow properties).

Both file formats may be used as a source for extracting the configuration data required for direct Spartan-6 User FPGA programming in Slave SelectMAP configuration mode.

Shown below are examples for the .bit file and the .bin file content (for configuration in SPI x4 mode). For direct User FPGA programming in Slave SelectMAP configuration mode, usually only the main configuration data area is used (the file header and the additional Spartan-6 configuration data is usually not programmed to the FPGA in Slave SelectMAP configuration mode).

For the .bit file example below, the data from address 0xAE on would be used for direct User FPGA programming in Slave SelectMAP configuration mode.

For the .bin file example below, the data from address 0x44 on would be used for direct User FPGA programming in Slave SelectMAP configuration mode.

.bit File Example

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  00 09 0F F0 0F F0 0F F0 0F F0 00 00 01 61 00 2B  File Header
00000010  74 70 6D 63 36 33 34 5F 78 63 36 73 6C 78 32 35
00000020  5F 65 78 61 2E 6E 63 64 3B 55 73 65 72 49 44 3D
00000030  30 78 46 46 46 46 46 46 46 46 00 62 00 0D 36 73
00000040  6C 78 32 35 66 67 67 34 38 34 00 63 00 0B 32 30
00000050  31 35 2F 30 32 2F 31 39 00 64 00 09 31 35 3A 32
00000060  35 3A 31 35 00 65 00 0C 3A FA FF FF FF FF FF FF  add. S6
00000070  FF FF FF FF FF FF FF FF FF FF AA 99 55 66 31 E1  Config Data
00000080  FF FF 32 61 00 44 32 81 6B 00 32 A1 00 44 32 C1
00000090  6B 00 32 E1 00 00 30 A1 00 00 33 01 31 00 32 01
000000A0  00 5F 30 A1 00 0E 20 00 20 00 20 00 20 00 FF FF
000000B0  FF FF FF FF FF FF FF FF FF FF FF FF FF FF AA 99  Config Data
000000C0  55 66 30 A1 00 07 20 00 31 A1 04 88 31 41 3D 08
000000D0  31 61 09 EE 31 C2 04 00 40 93 30 E1 00 CF 30 C1
000000E0  00 81 20 00 20 00 20 00 20 00 20 00 20 00 20 00
000000F0  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
…
000C3B40  00 17 C4 9B 30 A1 00 0D 20 00 20 00 20 00 20 00
000C3B50  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
000C3B60  20 00 20 00
```

.bin File Example

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  add. S6
00000010  AA 99 55 66 31 E1 FF FF 32 61 00 44 32 81 6B 00  Config Data
00000020  32 A1 00 44 32 C1 6B 00 32 E1 00 00 30 A1 00 00
00000030  33 01 31 00 32 01 00 5F 30 A1 00 0E 20 00 20 00
00000040  20 00 20 00 FF FF FF FF FF FF FF FF FF FF FF FF  Config Data
00000050  FF FF FF FF AA 99 55 66 30 A1 00 07 20 00 31 A1
00000060  04 88 31 41 3D 08 31 61 09 EE 31 C2 04 00 40 93
00000070  30 E1 00 CF 30 C1 00 81 20 00 20 00 20 00 20 00
00000080  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
…
000C3AD0  30 C1 00 81 30 02 00 17 C4 9B 30 A1 00 0D 20 00
000C3AE0  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
000C3AF0  20 00 20 00 20 00 20 00 20 00
```

7.3.2 Direct FPGA Programming

The registers used for direct (volatile) User FPGA In-System programming are located in the PCI Target Register Space.

Steps for direct (volatile) User FPGA Programming (Software point-of-view)

1. Write 0x00000003 to the User FPGA Configuration Control/Status Register (0xD0)

   This will assert the Spartan-6 FPGA PROGRAM_B pin and will prepare a FPGA re-configuration in Slave SelectMAP configuration mode.

   The Spartan-6 User FPGA will respond by driving both the DONE and INIT_B pins low.

2. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until bits 2 & 3 are clear

   I.e. wait until the Spartan-6 FPGA asserts the INIT_B and DONE pins low.

3. Write 0x00000001 to the User FPGA Configuration Control/Status Register (0xD0)

   This will release the Spartan-6 FPGA PROGRAM_B pin and will start a FPGA re-configuration in Slave SelectMAP configuration mode.

   The Spartan-6 FPGA will respond by releasing the INIT_B pin.

4. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until bit 3 is set

   I.e. wait until the Spartan-6 FPGA releases the INIT_B pin.

5. Write the configuration data 32 bit wise to the User FPGA Configuration Data Register (0xD4)

   Each register write loads four configuration data bytes to the Spartan-6 User FPGA.

   The register byte lane order send to the Spartan-6 User FPGA is: 1. REG[7:0], 2. REG[15:8], 3. REG[23:16], 4. REG[31:24].

   E.g. the Xilinx Spartan-6 Sync Word 0xAA995566 must be mapped to the register byte lanes (REG) as follows: 0xAA → REG[7:0], 0x99 → REG15:8], 0x55 → REG[23:16], 0x66 → REG[31:24].

6. Write eight more times (0xFFFFFFFF) to the User FPGA Configuration Data Register (0xD4)

   This will provide additional configuration clock cycles for the Spartan-6 FPGA configuration port which may be required (depending on the ISE project design and settings).

7. Wait for approx. 200ms

8. Read the User FPGA Configuration Control/Status Register (0xD0) and check if bit 2 is set

   I.e. check successful Spartan-6 FPGA configuration (DONE pin assertion).

9. Write eight more times (0xFFFFFFFF) to the User FPGA Configuration Data Register (0xD4)

   This will provide additional configuration clock cycles for the Spartan-6 FPGA configuration port which may be required (depending on the ISE project design and settings).

For each write to the User FPGA Configuration Data Register, four configuration clock cycles are automatically generated and the four register bytes are sent to the Spartan-6 User FPGA Slave SelectMAP configuration port (x8). The byte order sent to the Spartan-6 User FPGA configuration data port is (left to right): User FPGA Configuration Data Register bits [7:0], [15:8], [23:16], [31:24]. The User FPGA Configuration Data Register must be accessed with a 32 bit transfer size.

Since the Spartan-6 Slave SelectMAP configuration port (x8) expects the configuration data bytes to be bit-swapped, each byte is automatically bit-swapped by HW when send to the Spartan-6 User FPGA configuration port. Therefore, the source configuration data set must be non-bit-swapped. It is recommended to use the Xilinx .BIN file format as the configuration data source file. BIN file generation must be enabled in the Xilinx ISE design flow.

The required time for direct User FPGA configuration via PCI/Software is approx. 100ms.

7.4 FPGA Configuration via JTAG Header

The TPMC634 Xilinx Spartan-6 FPGA is configurable (volatile) via the TPMC634 JTAG header.

The Xilinx Spartan-6 FPGA always supports configuration via the JTAG port.

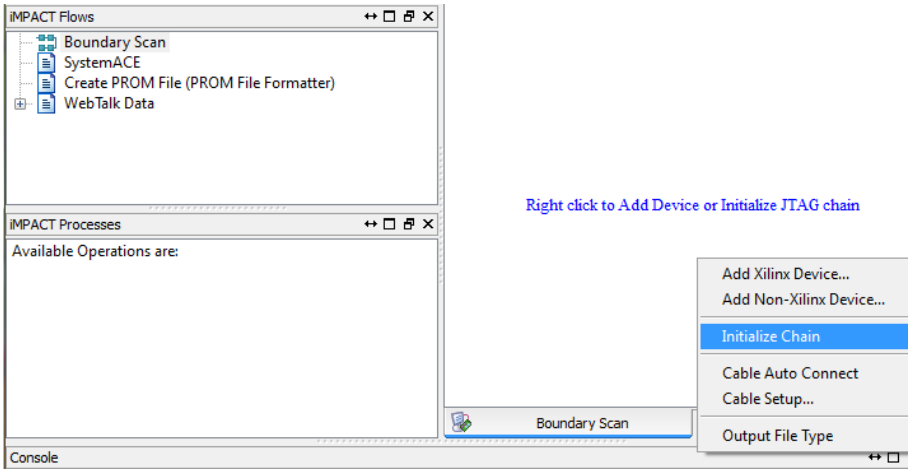
The Xilinx iMPACT software tool along with the Xilinx Platform Cable USB Programmer may be used for User FPGA configuration via the TPMC634 JTAG header.

The TPMC634 JTAG header directly supports the 14 pos. flat ribbon cable of the Xilinx Platform Cable USB Programmer.

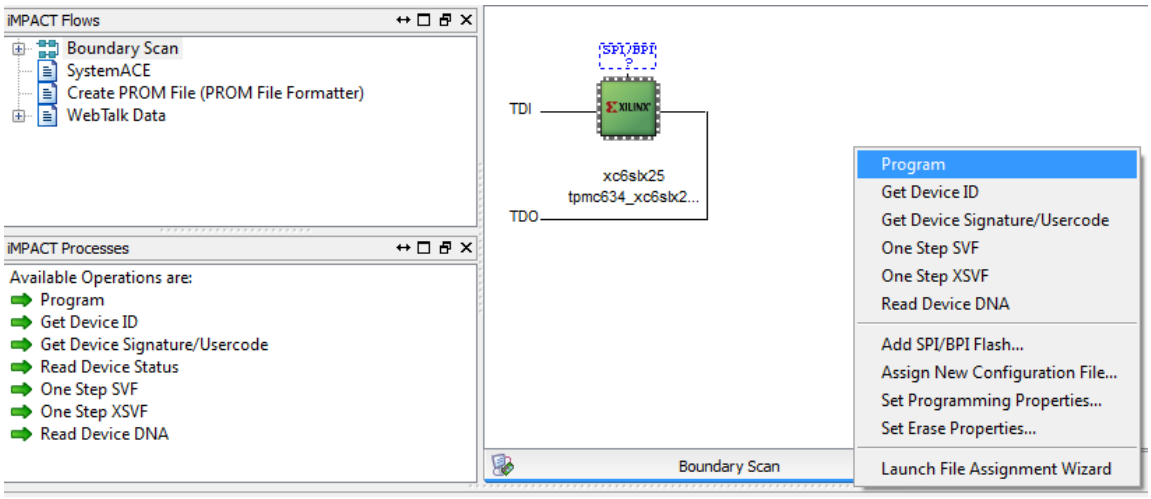
Steps for direct (volatile) User FPGA Configuration via the TPMC634 JTAG Header

1. Connect the Xilinx Platform Cable USB Programmer USB port to a PC
2. Connect the Xilinx Platform Cable USB Programmer JTAG port to the TPMC634 JTAG Header via the 14 pos. flat ribbon cable
3. Install the TPMC634 on a carrier and into a system
4. Turn on the system power supply

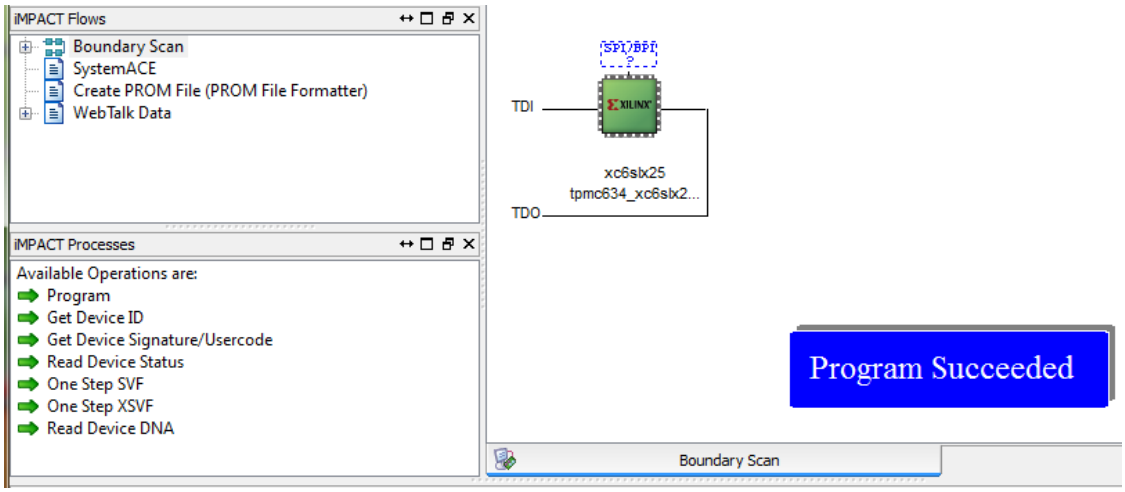
   Check: The status LED on the Xilinx Platform Cable USB Programmer should be on and green now.
5. Start the Xilinx iMPACT software
6. Select/Start the Boundary Scan iMPACT Flow
7. Right-Click and Initialize Chain



8. Check that a XC6SLX25 FPGA is detected as the only device in the JTAG chain
9. Assign the .bit File to the XC6SLX25 FPGA device. Do not attach an SPI or BPI PROM to the device.
10. Select (highlight) the XC6SLX25 FPGA device and select the Program operation



11. Check successful Program operation and FPGA configuration



12. Check that the TPMC634 S6 LED is on and green now → The TPMC634 Xilinx XC6SLX25 Spartan-6 User FPGA is now configured

8 SPI Flash Programming

8.1 SPI Flash Notes

The TPMC634 provides an on-board SPI Flash for storing the User FPGA configuration data.

The SPI Flash is the default User FPGA configuration option after power-up.

After Power-up, the Spartan-6 User FPGA always attempts to load the configuration bitstream from the on-board SPI Flash in *Master Serial / SPI* configuration mode.

Upon delivery, the SPI Flash contains a TPMC634 Spartan-6 User FPGA Example Application (a brief description is appended to this user manual).

8.1.1 SPI Flash Device Type

The TPMC634 uses a Winbond W25Q32FV (W25Q32BV compatible) 32Mbit serial SPI Flash with x4 mode support.

The TPMC634 SPI Flash Key Parameters are:

- 32Mbit (4Mbyte) Memory
- 16384 pages a 256 byte

  Page base addresses are 0x000000, 0x000100, 0x000200, …, 0x3FFF00
- 1024 sectors a 4096 byte (16 pages)

  Sector base addresses are 0x000000, 0x001000, 0x002000, …, 0x3FF000
- Address range 0h … 3FFFFFh

8.1.2 SPI Flash Programming Options

The TPMC634 provides the following options for programming the User FPGA SPI Flash:

- SPI Flash Programming via PCI/Software
- SPI Flash Programming by FPGA I/O Pins via JTAG Header

8.1.3 SPI Flash Non-Volatile QE Bit

The SPI Flash device provides an internal non-volatile Quad-Enable (QE) bit that must be set for using SPI x4 (quad) mode during FPGA configuration from the SPI Flash in Master Serial/SPI configuration mode.

The TPMC634 comes with an example code stored in the SPI Flash build for x4 SPI mode and with the non-volatile QE bit programmed as set.

Since the QE bit is a non-volatile bit, it will remain as programmed for subsequent power cycles. The TPMC634 SPI Flash In-System-Programming instructions (Page Program, Page Read, Chip Erase and Sector Erase) will not alter the SPI Flash QE bit setting.

However, when connected to the TPMC634 JTAG header, third party tools like the Xilinx iMPACT software may clear the non-volatile QE bit during certain operations. Typically the Xilinx iMPACT software will clear the non-volatile QE bit during an SPI Flash Erase operation and will set the non-volatile QE bit again during

an SPI Flash Program operation only if SPI x4 mode has been selected in the Xilinx iMPACT flow configuration. So due to certain Xilinx iMPACT software operations, the non-volatile QE bit may get cleared, and once cleared, will stay so for subsequent power cycles.

If the QE bit ever should get cleared by third party tools, measures must be taken to set the bit again in case SPI x4 mode is planned to be used for FPGA configuration from the SPI Flash in Master/Serial SPI configuration mode, especially if the new project configuration data is written to the SPI Flash via the PCI bus.

The TPMC634 provides an In-System-Programming instruction for setting the non-volatile SPI Flash QE bit by software.

8.1.4 Xilinx ISE / BitGen Options

In the Xilinx ISE Design Flow of the User FPGA Project, use the following settings in the *Generate Programming File* process properties:

- General Options
  - –g Binary (Create Binary Configuration File): Yes
- Configuration Options
  - –g ConfigRate: 12
  - –g ExtMasterCclk_en: No
  - –g SPI_buswidth: 4

Note that the non-volatile QE bit in the SPI Flash must be set for FPGA configuration in SPI x4 mode.

8.2 SPI Flash Programming via PCI/Software

The TPMC634 supports In-System SPI Flash Programming in x1 mode. The configuration data that is programmed into the SPI Flash may be for configuration in SPI x1 mode or SPI x4 mode. It is recommended to generate the configuration data for configuration in SPI x4 mode.

PCI BAR 0 (PCI Target Register Space) and PCI BAR 1 (In-System Programming Space) are used for SPI Flash In-System Programming.

SPI Flash programming via PCI/Software requires the ISP_EN bit in the ISP Control Register (0xE0) to be set. It is recommended to clear the ISP_EN bit when the SPI Flash programming is done.

8.2.1 SPI Flash Program Data

The .bin file from the Xilinx ISE design flow is recommended to be used as the data source for SPI Flash programming.

The .bin file generation must be enabled by a BitGen option (or in the ISE design flow *Generate Programming File* process properties).

A .bin file example for configuration in SPI x4 mode is shown below.

Both the additional Spartan-6 configuration data and the main configuration data must be programmed into the SPI Flash page-wise via the ISP Space (PCI BAR 1) and the SPI Flash *Page Program* instruction.

During the SPI Flash page program instruction, the bytes are transferred to the SPI Flash from lower to higher ISP Space (PCI BAR 1) byte addresses.

.bin File Example (for configuration in SPI x4 mode)

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F

00000000  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF  add. S6
00000010  AA 99 55 66 31 E1 FF FF 32 61 00 44 32 81 6B 00  Config Data
00000020  32 A1 00 44 32 C1 6B 00 32 E1 00 00 30 A1 00 00
00000030  33 01 31 00 32 01 00 5F 30 A1 00 0E 20 00 20 00
00000040  20 00 20 00 FF FF FF FF FF FF FF FF FF FF FF FF  Config Data
00000050  FF FF FF FF AA 99 55 66 30 A1 00 07 20 00 31 A1
00000060  04 88 31 41 3D 08 31 61 09 EE 31 C2 04 00 40 93
00000070  30 E1 00 CF 30 C1 00 81 20 00 20 00 20 00 20 00
00000080  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
…
000C3AD0  30 C1 00 81 30 02 00 17 C4 9B 30 A1 00 0D 20 00
000C3AE0  20 00 20 00 20 00 20 00 20 00 20 00 20 00 20 00
000C3AF0  20 00 20 00 20 00 20 00 20 00
```

8.2.2 SPI Flash Instructions

The following SPI Flash instructions are supported:

- Chip Erase (0x60)
- Sector Erase (0x20)
- Page Program (program a 256 Byte SPI Flash page) (0x02)
- Page Read (read a 256 Byte SPI Flash page) (0x03)
- Set non-volatile QE Bit (0x31)

Program and Read instructions are processed on a per-page base (page-by-page).

8.2.3 Steps for SPI Flash Chip Erase Operation

1. Write 0x00000001 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the PCI Target Device is connected to the SPI Flash.

2. Configure the SPI Flash *Chip Erase* instruction in the ISP Configuration Register (0xE4)

   I.e. write 0x00000060 to the ISP Configuration Register (0xE4).

3. Write 0x00000001 to the ISP Command Register (0xE8)

   This will start the configured SPI Flash instruction.

4. Keep reading the ISP Status Register (0xEC) until bit 1 is read as 0

   I.e. wait until the SPI Flash instruction is no longer busy.

   Alternatively, instead of polling, an interrupt scheme may be used.

5. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash (when in Master Serial / SPI configuration mode).

8.2.4 Steps for SPI Flash Sector Erase Operation

1. Write 0x00000001 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the PCI Target Device is connected to the SPI Flash.

2. Configure the SPI Flash *Sector Erase* instruction in the ISP Configuration Register (0xE4)

   The sector base address and the sector erase instruction code must be set accordingly. The sector erase instruction code is 0x20.

   For erasing the first SPI Flash sector (base address 0x000000) write 0x00000020 to the ISP Configuration Register (0xE4).

   For erasing the second SPI Flash sector (base address 0x001000) write 0x00100020 to the ISP Configuration Register (0xE4).

   For erasing the third SPI Flash sector (base address 0x002000) write 0x00200020 to the ISP Configuration Register (0xE4).

   …

   For erasing the last SPI Flash sector (base address 0x3FF000) write 0x00F03F20 to the ISP Configuration Register (0xE4).

3. Write 0x00000001 to the ISP Command Register (0xE8)

   This will start the configured SPI Flash instruction.

4. Keep reading the ISP Status Register (0xEC) until bit 1 is read as 0

   I.e. wait until the SPI Flash Instruction is no longer busy.

   Alternatively, instead of polling, an interrupt scheme may be used.

5. If not done, go to 2 for erasing the next SPI Flash sector

6. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash (when in Master Serial / SPI configuration mode).

8.2.5 Steps for SPI Flash Program Operation

Note that any SPI Flash page that is to be programmed must be in an erased state before programming.

1. Write 0x00000001 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the PCI Target Device is connected to the SPI Flash.

2. Write the Page Program data to the ISP Space (PCI BAR 1)

   During the SPI Flash page program instruction, the bytes are transferred to the SPI Flash from lower to higher byte addresses.

3. Configure the SPI Flash *Page Program* instruction in the ISP Configuration Register (0xE4)

   The page base address and the page program instruction code must be set accordingly. The page program instruction code is 0x02.

   For programming the first SPI Flash page (base address 0x000000) write 0x00000002 to the ISP Configuration Register (0xE4).

   For programming the second SPI Flash page (base address 0x000100) write 0x00010002 to the ISP Configuration Register (0xE4).

   For programming the third SPI Flash page (base address 0x000200) write 0x00020002 to the ISP Configuration Register (0xE4).

   …

   For programming the last SPI Flash page (base address 0x3FFF00) write 0x00FF3F02 to the ISP Configuration Register (0xE4).

4. Keep reading the ISP Status Register (0xEC) until bit 1 is read as 0

   I.e. wait until a former SPI Flash instruction is no longer busy.

5. Write 0x00000001 to the ISP Command Register (0xE8)

   This will start the configured SPI Flash instruction.

6. Keep reading the ISP Status Register (0xEC) until bit 0 is read as 0

   I.e. wait until the page data has been transferred from the ISP Data Space to the SPI Flash.

   The SPI Flash internal programming sequence may not be done yet, but any next page data may now be written to the ISP Data Space.

7. If not done, go to 2 for programming the next SPI Flash page

8. Keep reading the ISP Status Register until bit 1 is read as 0

   I.e. wait until the SPI Flash instruction is no longer busy.

9. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash (when in Master Serial / SPI configuration mode).

8.2.6 Steps for SPI Flash Read Operation

1. Write 0x00000001 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the PCI Target Device is connected to the SPI Flash.

2. Configure the SPI Flash *Page Read* instruction in the ISP Configuration Register (0xE4)

   The page base address and the page read instruction code must be set accordingly. The page read instruction code is 0x03.

   For reading the first SPI Flash page (base address 0x000000) write 0x00000003 to the ISP Configuration Register (0xE4).

For reading the second SPI Flash page (base address 0x000100) write 0x00010003 to the ISP Configuration Register (0xE4).

For reading the third SPI Flash page (base address 0x000200) write 0x00020003 to the ISP Configuration Register (0xE4).

…

For reading the last SPI Flash page (base address 0x3FFF00) write 0x00FF3F03 to the ISP Configuration Register (0xE4).

3. Write 0x00000001 to the ISP Command Register (0xE8)

   This will start the configured SPI Flash instruction.

4. Keep reading the ISP Status Register (0xEC) until bit 1 is read as 0

   I.e. wait until the SPI Flash Instruction is no longer busy.

   Alternatively, instead of polling, an interrupt scheme may be used.

5. Copy the SPI Flash page data from the ISP Space (PCI BAR 1)

6. If not done, go to 2 for reading the next SPI Flash page

7. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash (when in Master Serial / SPI configuration mode).

8.2.7 Steps for setting the SPI Flash Non-Volatile QE Bit

1. Write 0x00000001 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the PCI Target Device is connected to the SPI Flash.

2. Configure the SPI Flash *Set QE Bit* instruction in the ISP Configuration Register (0xE4)

   I.e. write 0x00000031 to the ISP Configuration Register (0xE4).

3. Write 0x00000001 to the ISP Command Register (0xE8)

   This will start the configured SPI Flash instruction.

4. Keep reading the ISP Status Register (0xEC) until bit 1 is read as 0

   I.e. wait until the SPI Flash instruction is no longer busy.

5. Write 0x00000000 to the ISP Control Register (0xE0)

   This will set the on-board configuration signal path so that the Spartan-6 User FPGA is connected to the SPI Flash (when in Master Serial / SPI configuration mode).

8.3 SPI Flash Programming via JTAG Header

Programming the SPI Flash via the TPMC634 JTAG header requires the Xilinx iMPACT software and the Xilinx Platform Cable USB Programmer with the 2mm pitch 14 pos. flat ribbon cable.

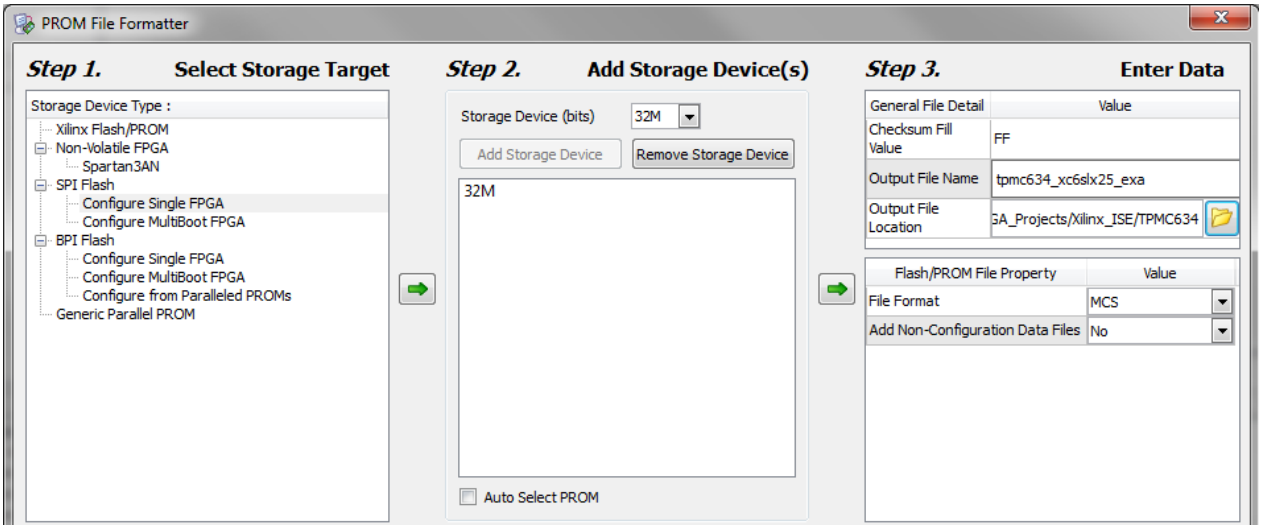
The Xilinx iMPACT software supports indirect SPI Flash programming by loading (and running) a special SPI core logic into the Spartan-6 FPGA that stimulates the FPGA I/O pins connected to the SPI Flash as required by the SPI programming protocol. The current User FPGA configuration will be overwritten by the SPI core logic.

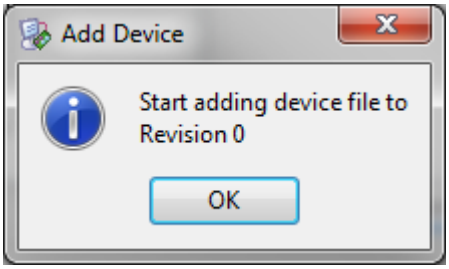
Programming the SPI Flash via the TPMC634 JTAG header requires the ISP_EN bit in the ISP Control Register (0xE0) to be clear. Furthermore the Master Serial / SPI configuration mode must be set for the User FPGA. These are power-up default conditions.

8.3.1 MCS Programming File Generation

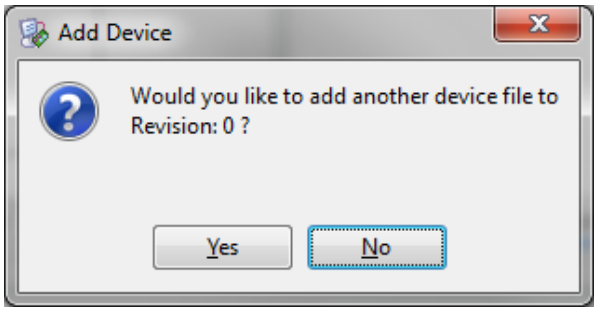
For programming the SPI Flash via the TPMC634 JTAG Header, the resulting .BIT file from the Xilinx ISE Design Flow is converted to an appropriate .MCS programming file first by using the Xilinx iMPACT Prom File Formatter Flow as shown below.

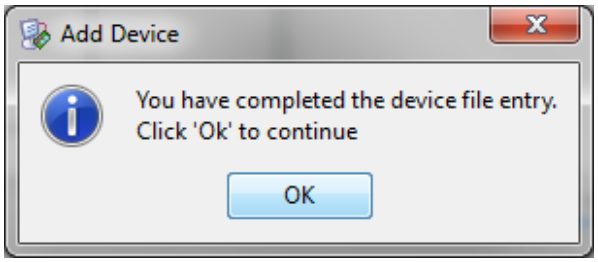
- PROM File Formatter Steps 1 – 3
    - Step 1 (Select Storage Target) → Select SPI Flash / Configure Single FPGA
    - Step 2 (Add Storage Device) → Add a 32Mbit Storage Device
    - Step 3 (Enter Data) → Select .MCS File Format, Output File Name & Location



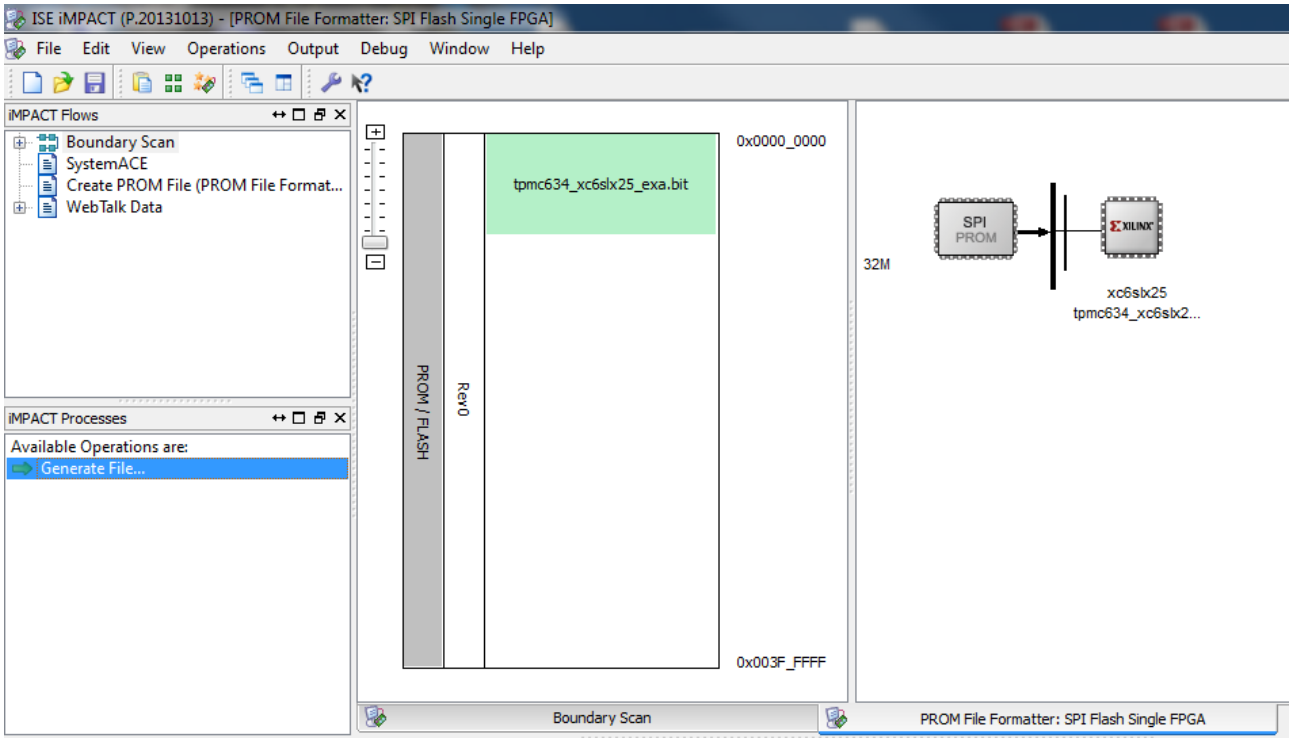


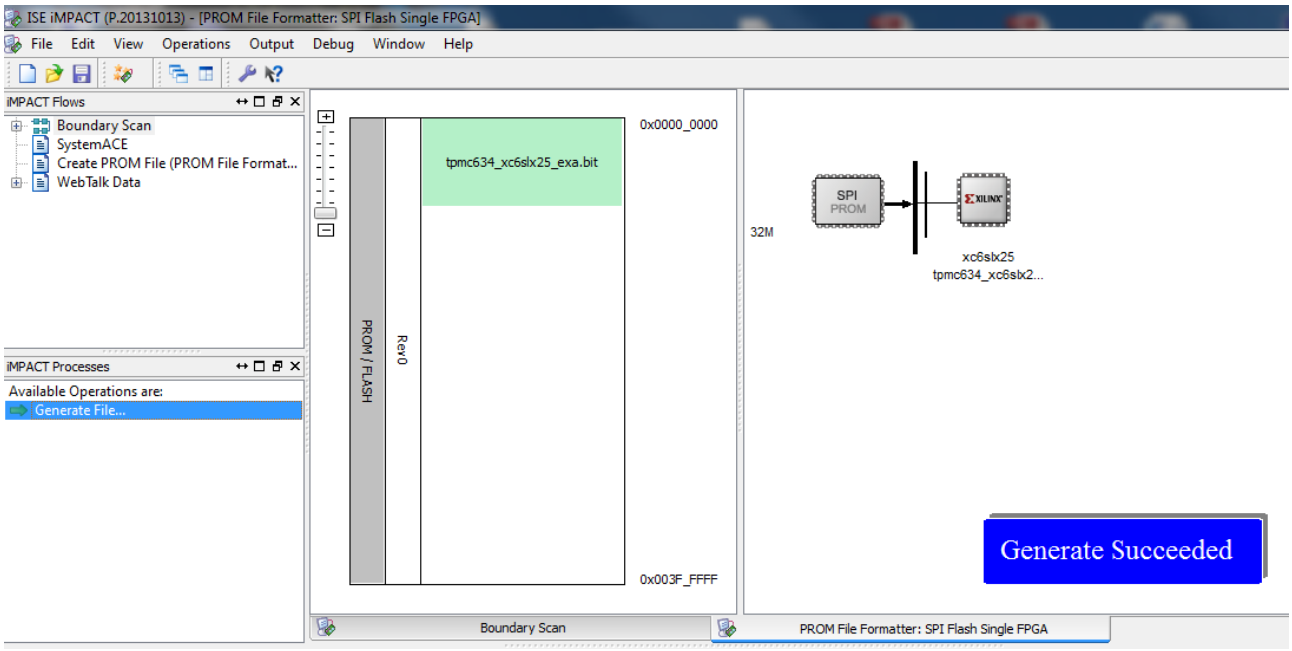
- Add the resulting .BIT file from the Xilinx ISE Design Flow to represent the XC6SLX25 FPGA device





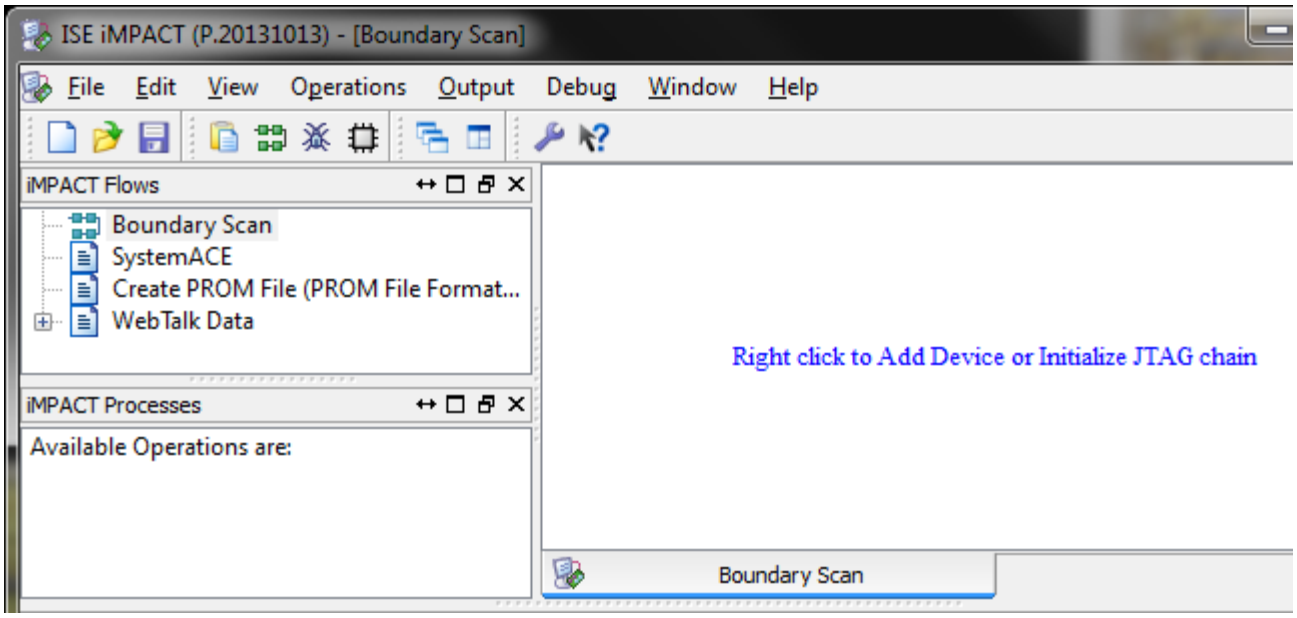
- Generate the .MCS file



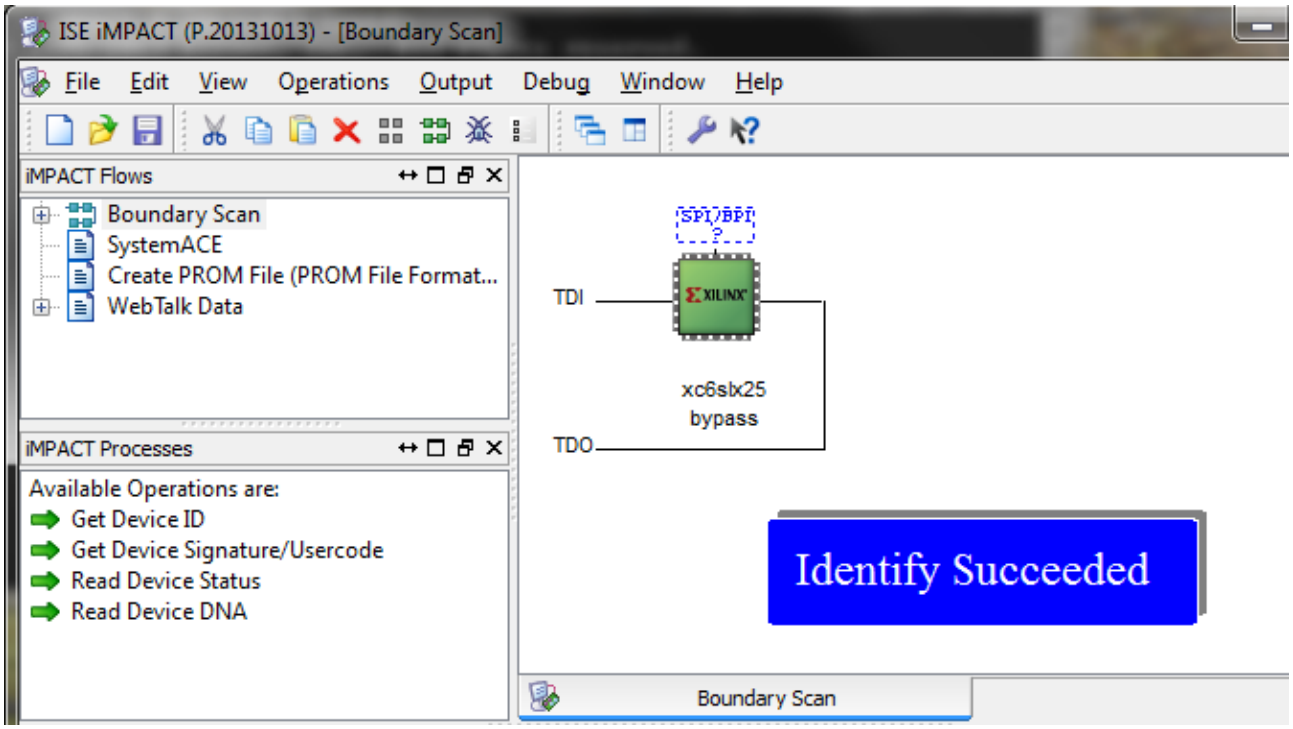


8.3.2 SPI Flash Programming

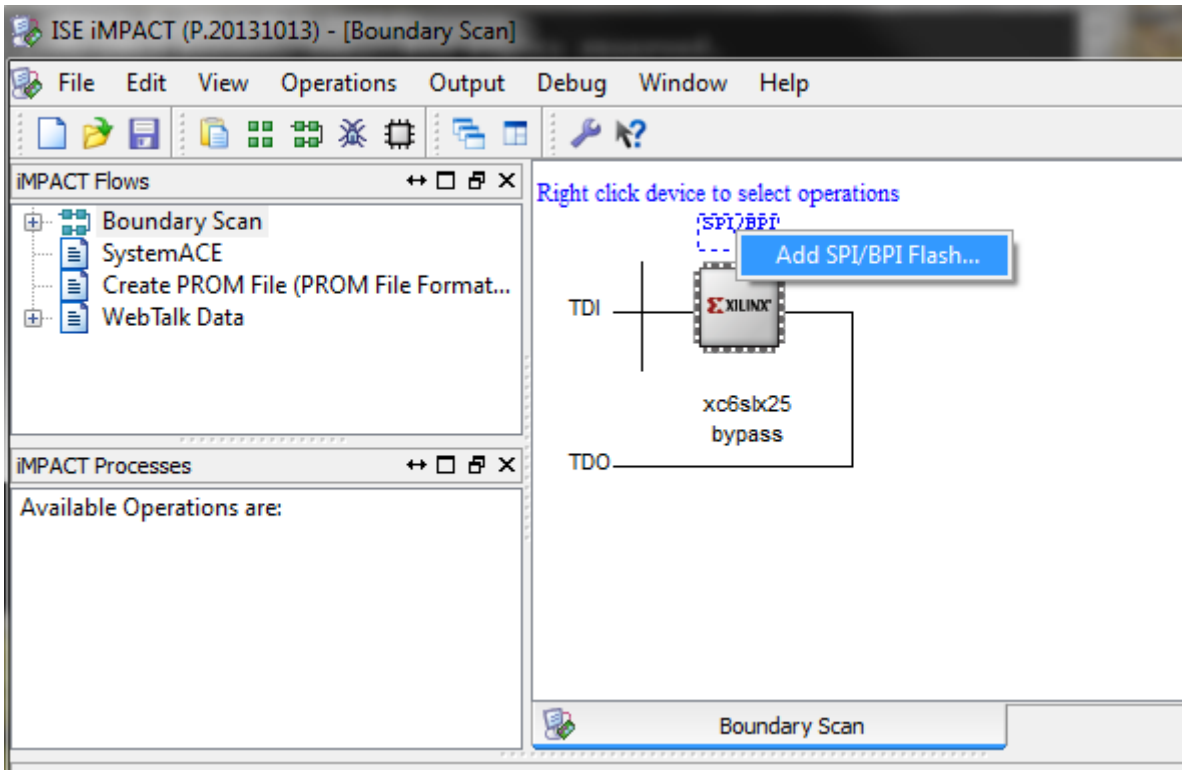
- Prepare a system and an appropriate PMC carrier
- Connect the Xilinx Platform Cable USB Programmer USB port to the PC/Notebook
    - Check: The status LED on the Xilinx Platform Cable USB Programmer should be lit orange now
- Connect the Xilinx Platform Cable USB Programmer JTAG port via the 14 pos. flat ribbon cable to the TPMC634 JTAG header
- Install the TPMC634 on the PMC carrier, install the PMC carrier into the system and turn on the system
    - Check: The status LED on the Xilinx Platform Cable USB Programmer should turn green now
    - Check: The TPMC634 power good (PG) and MachXO2 (XO) LED should be lit green now
- Start the Xilinx iMPACT program
- Select the Boundary Scan iMPACT Flow
- Initialize the JTAG chain



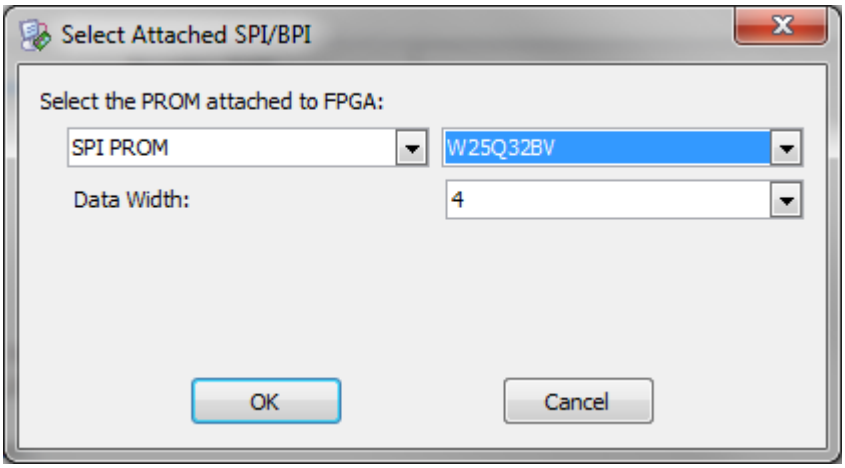
- Verify the auto-detected JTAG chain (see figure below) and set the XC6SLX25 FPGA device to BYPASS



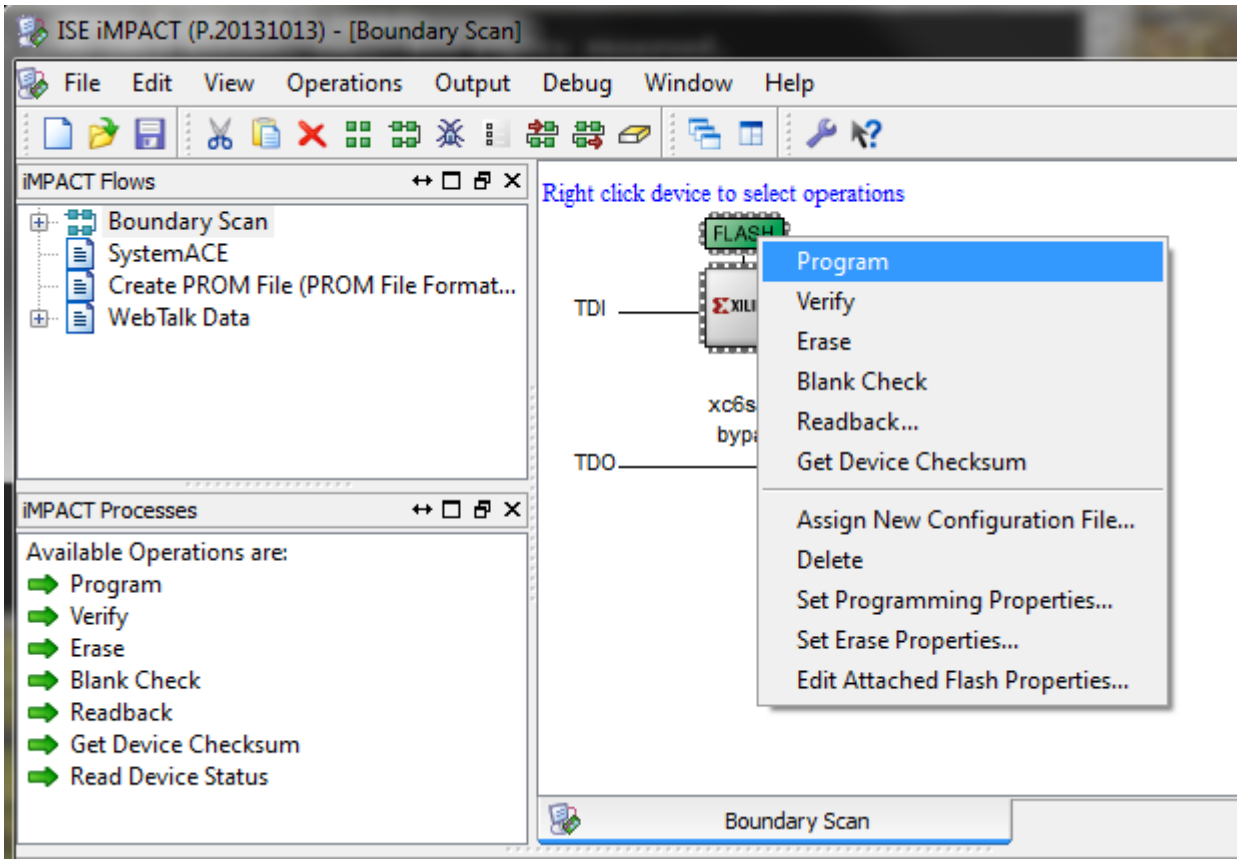
- Assign the prepared .MCS file to the SPI Flash
    - Right-Click on the SPI Flash and select Add SPI/BPI Flash

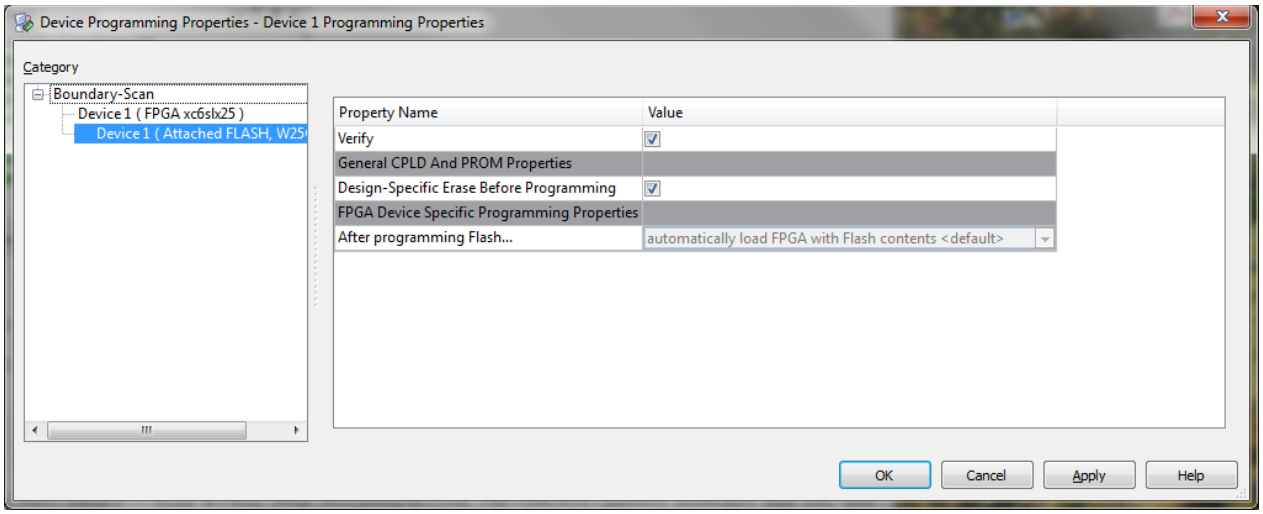


- Configure the SPI Flash Device
  - Select the W25Q32BV as SPI PROM
  - Data Width depends on the chosen SPI_buswidth BitGen option during the Xilinx ISE Design Flow

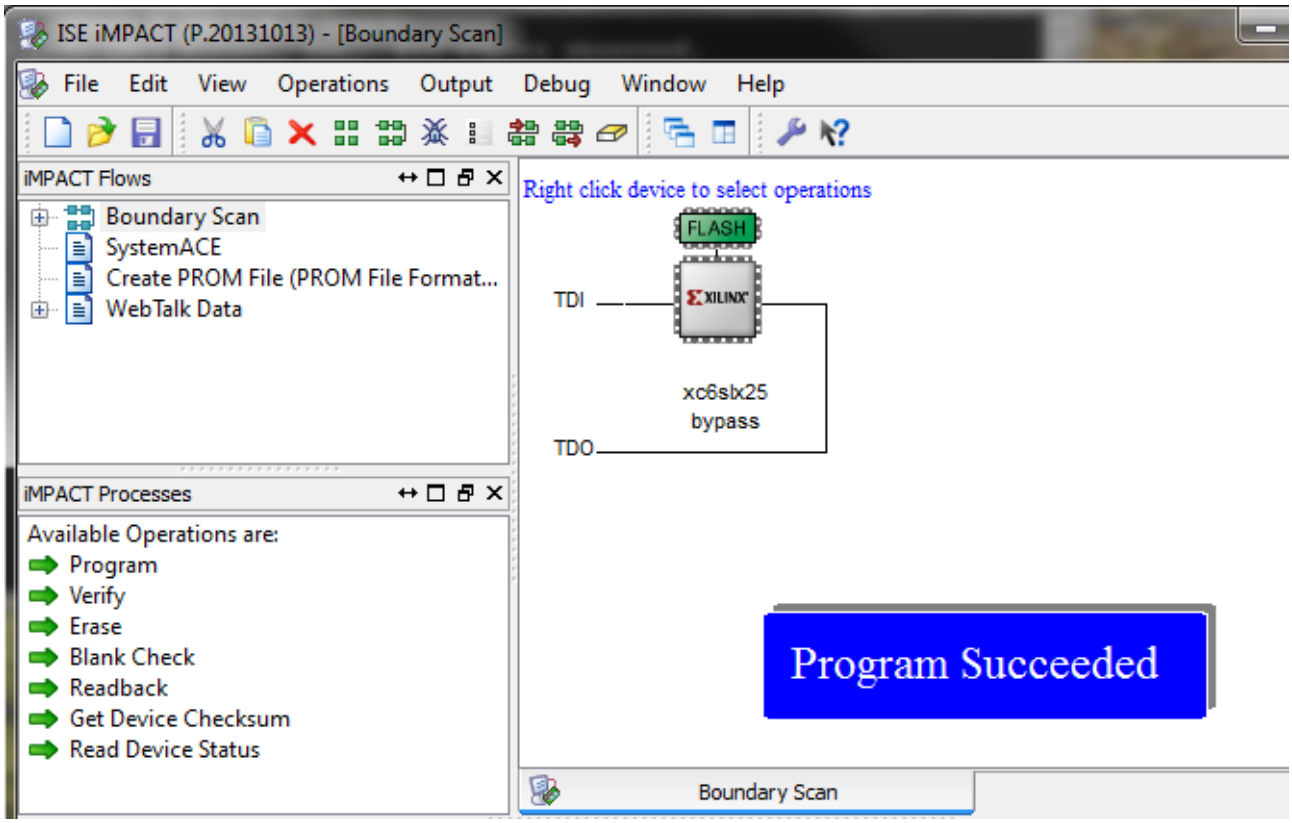


- Program the SPI Flash
  - Right-Click on the SPI Flash and select *Program*
  - Select the *Verify* and *Design-Specific Erase Before Programming* check options





- Start SPI Flash programming



- Wait and Check successful SPI Flash programming
- Turn the system off and unplug the Xilinx Platform Cable USB I/II JTAG port from the TPMC634
- Turn the system on again
    - Check: The three TPMC634 LEDs must be lit green now (PG: Power Good, XO: MachXO2 Operating, S6: User FPGA Configured)

9 Local Bus Interface

9.1 Local Bus Interface Notes

The TPMC634 provides an on-board off-chip local bus for accessing the User FPGA as a target.

For each optional user space (PCI BARs 2 to 5) there is a dedicated user space select signal.

The TPMC634 local bus is synchronous and address/data multiplexed.

The local bus clock is a buffered version of the PCI clock signal. The actual local bus clock frequency depends on the actual PCI clock frequency (8 MHZ minimum, 33 MHz maximum).

The local bus interface is a 32-bit port. Address lines LB_AD[1:0] are “00” during the address phase.

Lower address lines used for decoding within the space size range are passed from the PCI bus to the local bus. Upper address lines used for space base decoding are passed as 0.

PCI data byte lanes are mapped 1:1 to the local bus data byte lanes, e.g. PCI_AD[7:0] is mapped to LB_AD[7:0] etc. Each user space is operating in little endian mode.

A PCI reset is passed to the local bus reset signal. The local bus reset is also held active while the User FPGA is un-configured, while the local bus reset control bit is set in the Local Bus Interface Register or while the Local Bus Clock PLL is not in the lock state.

9.2 Local Bus Cycle Description

For a local bus cycle the MachXO2 PCI target chip is acting as the local bus master while the Spartan-6 User FPGA is acting as a local bus target.

During/throughout a local bus cycle the master asserts a cycle signal, a direction signal and a user space select signal. When the cycle signal is not active, the local bus is considered to be in an idle phase.

Regarding the multiplexed address/data bus, a local bus cycle consists of an address phase (for passing the address to the target) and a data phase (for passing write or read data between master and target). During a local bus cycle the master indicates the type of the current local bus cycle phase with a dedicated address/data phase signal. The master provides valid byte enable signals during/throughout the data phase.

Both the master and the target are providing a local bus cycle phase ready signal. Local bus cycle phases are done when both the master and the target are indicating phase ready status.

During the address phase the master ready signal indicates a valid address on the multiplexed address/data bus. For the address phase the target ready signal indicates that the target is ready to sample the address from the multiplexed address/data bus.

The address phase is done when both the master and the target are indicating ready status.

During a write cycle data phase, the master ready signal indicates valid write data on the multiplexed address/data bus and the target ready signal indicates that the target is ready to sample the write data from the multiplexed address/data bus.

During a read cycle data phase, the target ready signal indicates valid read data on the multiplexed address/data bus and the master ready signal indicates that the master is ready to sample the read data from the multiplexed address/data bus.

The data phase is done when both master and target are indicating ready status.

9.2.1 Local Bus Master Abort (Local Bus Time-Out)

The TPMC634 provides a local bus time-out monitor. The local bus time-out function is enabled by default and could be disabled in the Local Bus Interface Register.

When the local bus time-out function is enabled, an internal timer is started at the beginning of a local bus cycle. When the timer expires during the local bus cycle (64 local clock cycles), a time-out event occurs. When a local bus time-out occurs, the master indicates the event to the target on a dedicated signal line (for a single clock cycle) and the local bus cycle terminates. The event is stored in a flag in the Local Bus Interface Register and is also capable of generating an interrupt. A (posted) write local bus time-out is not reported via the PCI bus. A (delayed) read local bus time-out is reported on the PCI bus as a PCI target abort returning 0xFFFFFFFF as read data.

When the local bus time-out feature is disabled, the master will wait for the target to indicate ready (or error) status during a local bus cycle. When the target fails to indicate ready (or error) status on the local bus, the local bus cycle will hang. As a consequence a PCI delayed read command would show permanent retries on the PCI bus, causing the PCI bus to hang. PCI posted writes would fill-up the PCI command FIFO, causing the PCI bus to hang after a short time. This mode may be used for extremely slow local bus target designs or during a SW development/test phase.

Note that if the local bus time-out function is enabled, a local bus time-out will occur when the PCI master initiates a local bus access while the User FPGA is not configured accordingly.

Note that if the local bus time-out function is disabled, the Local & PCI bus will hang (immediately for reads) when the PCI master initiates a local bus access while the User FPGA is not configured accordingly.

9.2.2 Local Bus Target Error

The TPMC634 local bus design provides a signal for indicating target error conditions to the master during local bus cycles.

Possible target error causes may be:

- byte enable mismatch
- invalid address
- invalid / unsupported selected space

Note that the target error causes are implemented by the user FPGA logic.

If generated by the target, the signal must be generated during a local bus cycle for a single clock cycle.

Upon detecting a target error the current local bus cycle will be terminated by the master and the affected local bus read/write command is discarded.

A local bus target error event is stored in a flag in the Local Bus Interface Register. The event is capable of generating an interrupt but is not reported via the PCI bus in any other way.

9.3 Local Bus Signal Description

Local Bus Signal Description		
Signal	**Direction**	**Description**
LB_CLK	Input for M & T	Local Bus Clock A buffered version of the PCI clock signal (max. 33 MHz)
LB_RST#	M → T	Local Bus Reset 0: Reset State 1: Operating State Local Bus Reset sources are: • PCI reset • User FPGA not configured • Local Bus SW reset, • Local Clock PLL not locked
LB_AD[31:0]	M ←→ T	Local Bus Address/Data Bus (Multiplexed) Only valid during an active local bus cycle **Address Phase:** Master drives address on the AD bus **Data Phase:** Writes: Master drives write data on the AD bus Reads: Target drives read data on the AD bus
LB_CYC#	M → T	Local Bus Cycle Validates a local bus cycle 0: Local Bus Cycle Active (Address or Data Phase) 1: Local Bus in Idle Phase
LB_SEL#[3:0]	M → T	Local Bus User Space Select Same timing as LB_CYC# Indicates the selected user space (active low) Only one user space is selected for a local bus cycle PCI BAR 2 <> User Space 0 PCI BAR 3 <> User Space 1 PCI BAR 4 <> User Space 2 PCI BAR 5 <> User Space 3
LB_R/W#	M → T	Local Bus Direction Same timing as LB_CYC# Only valid during an active local bus cycle 0: Write Cycle 1: Read Cycle
LB_BE#[3:0]	M → T	Local Bus Byte Enables Only valid during the data phase of a local bus cycle Indicates active byte lanes on the AD bus (active low) Either one, two or four byte lanes are active during the data phase of a local bus cycle

Local Bus Signal Description		
Signal	**Direction**	**Description**
		Byte Lane 0 <> AD[7:0] Byte Lane 1 <> AD[15:8] Byte Lane 2 <> AD[23:16] Byte Lane 3 <> AD[31:24]
LB_D/A#	M → T	Local Bus Data / Address Phase Only valid during an active local bus cycle 0: Address Phase 1: Data Phase
LB_MRDY#	M → T	Local Bus Master Ready Only valid during an active local bus cycle **Address Phase:** 0: Master drives valid Address on AD bus 1: Address on AD bus is not valid **Data Phase:** Writes: 0: Master drives valid Write Data on AD bus 1: Write Data on AD bus is not valid Reads: 0: Master is ready to sample Read Data from the AD bus. 1: Master is not ready to sample Read Data from the AD bus.
LB_MABT#	M → T	Local Bus Master Abort The master may abort a local bus cycle because of a time-out condition (no target ready or target error response within expected time). In case of a master abort, the master will drive this signal low for one clock cycle and will terminate and discard the current local bus cycle.
LB_TRDY#	M ← T	Local Bus Target Ready Only valid during an active local bus cycle **Address Phase:** 0: Target is ready to sample the Address from the AD bus 1: Target is not ready to sample the Address from the AD bus **Data Phase:** Writes: 0: Target is ready to sample the Write Data from the AD bus 1: Target is not ready to sample the Write Data from the AD bus Reads: 0: Target drives valid Read Data on the AD bus 1: Read Data on AD bus is not valid
LB_TERR#	M ← T	Local Bus Target Error The target may report an error condition during a local bus

Local Bus Signal Description		
Signal	**Direction**	**Description**
		cycle by driving this signal low for one local clock cycle. In case of a target error indication during a local bus cycle, the master will terminate and discard the local bus cycle.
LB_INT#	M ← T	Local Bus Target Interrupt The target may assert a level sensitive interrupt by driving this signal low This signal has an on-board pull-up resistor

M = Master, T = Target

Table 9-1 : Local Bus Signal Description

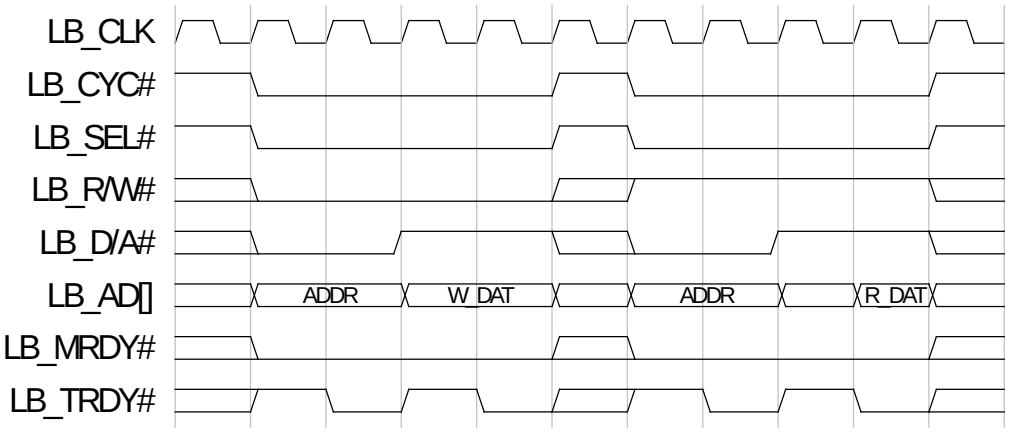
9.4 Local Bus Interface Timing

Local Bus Interface Timing Parameter	Time Value
Local Bus Clock Period Time	
LB_CLK	>= 30.3 ns
Local Bus Master Input Setup Timing	
LB_AD[31:0]	5 ns
LB_TRDY#, LB_TERR#	10 ns
Local Bus Master Output Delay Timing	
LB_AD[31:0]	11 ns
LB_SEL#[3:0], LB_BE#[3:0], LB_CYC#, LB_R/W#, LB_D/A#, LB_MRDY#, LB_MABT#	11 ns
Recommended Local Bus Target Input Setup Timing	
LB_AD[31:0]	10 ns
LB_SEL#[3:0], LB_BE#[3:0], LB_CYC#, LB_R/W#, LB_D/A#, LB_MRDY#, LB_MABT#	10 ns
Recommended Local Bus Target Output Delay Timing	
LB_AD[31:0]	15 ns
LB_TRDY#, LB_TERR#	12 ns

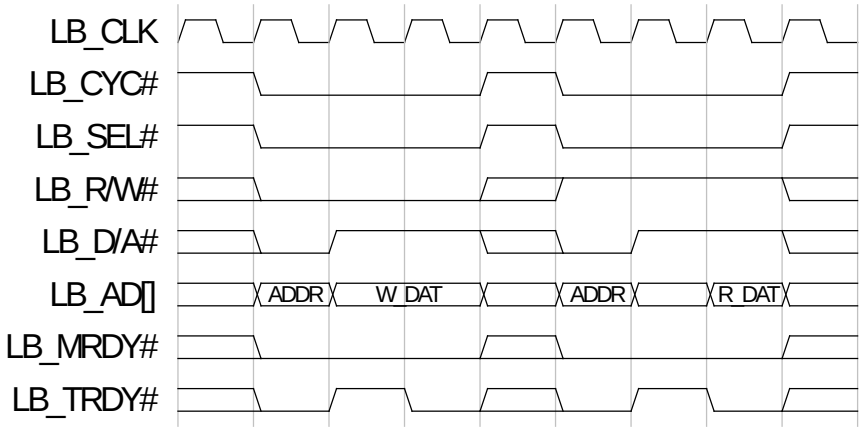
Table 9-2 : Local Bus Interface Timing

9.5 Local Bus Signal Protocol Example Diagrams

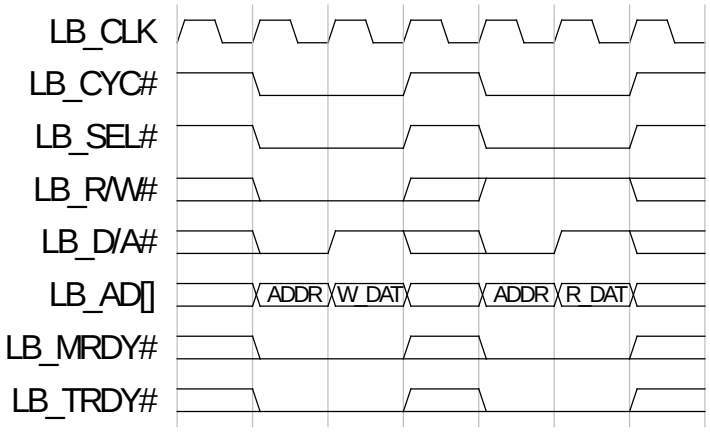
The following figures are showing some examples for the local bus signal protocol.



Fast Master Slow Target Local Bus Example



Fast Master Moderate Target Local Bus Example



Fast Master Fast Target Local Bus Example

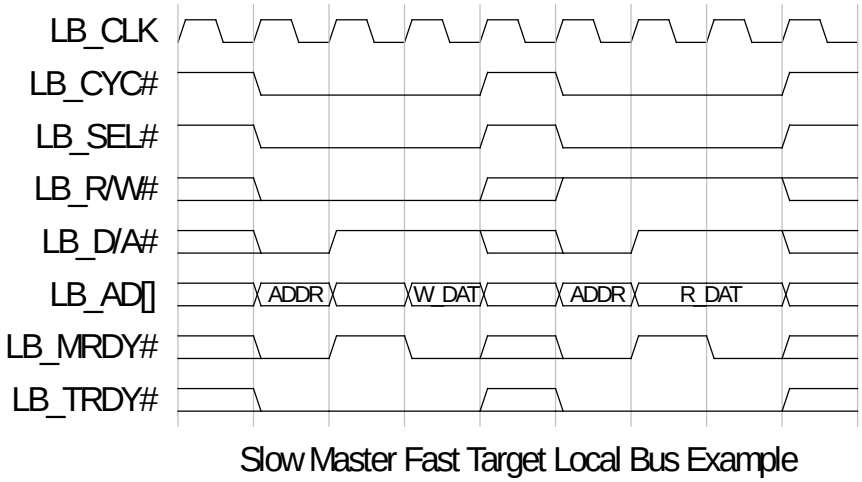


Figure 9-1 : Local Bus Signal Protocol Example Diagrams

(1) Note that the following signals are only valid during an active local bus cycle:

LB_R/W#, LB_D/A#, LB_MRDY#, LB_TRDY#, LB_MABT#, LB_TERR#

(2) Note that (though shown in all examples above) the master may not necessarily be ready with a valid address in the very first clock cycle of the local bus cycle.
A valid address is indicated by LB_CYC# = 0 and LB_D/A# = 0 and LB_MRDY# = 0.

(3) Note that there is no fix local bus signal protocol configuration. Any other local bus cycle may show a different signal/time protocol.

10 Interrupts

10.1 Interrupt Sources

On the TPMC634 there are multiple interrupt sources capable of generating a PCI INTA# interrupt.

The TPMC634 provides the following interrupt sources:

- SPI Flash In-System Programming Interrupts

  There are two interrupts sources available for SPI Flash In-System Programming.

  One interrupt is for indicating that the SPI page data (PCI BAR 1) processing is done (so page read data may be read or next page write data may be written).

  Another interrupt is for indicating that an SPI instruction is done (so another one may be started).

- Local Bus User Interrupt

  A dedicated User FPGA I/O (output) pin is designated as the local bus user interrupt signal.

- Local Bus Error Interrupt

  There is one interrupt source indicating local bus error events.

  There are three local bus related error events capable of generating the local bus error interrupt.

  - Local Bus Master Abort (Time-Out)
  - Local Bus Target Error
  - Local Bus Clock PLL Loss-of-Lock

10.2 Interrupt Handling

The PCI Target Register Space provides an interrupt enable register, an interrupt status register and an interrupt configuration register for interrupt handling.

Interrupts are enabled in the interrupt enable register. If enabled, upon an interrupt event the corresponding bit in the interrupt status register is set. If any interrupt status bit is set in the interrupt status register, an interrupt request is asserted on the PIC INTA# line.

Interrupt clearing depends on the interrupt acknowledge mode configured in the interrupt configuration register. There are two interrupt acknowledge modes. In ‘clear-by-write’ mode, interrupt status bits are cleared by writing a ‘1’ to the corresponding bit in the interrupt status register. In ‘clear-on-read’ mode, interrupt status bits are automatically cleared by reading the interrupt status register. Disabling interrupts also clears the corresponding bits in the interrupt status register (thus clears a corresponding interrupt).

The User FPGA logic design may also implement an interrupt status register in case of multiple User FPGA internal local bus user interrupt sources.

10.2.1 SPI Flash In-System Programming Interrupts

ISP SPI Instruction Done Interrupt

If enabled, this event based interrupt becomes active when the ISP SPI Instruction Busy status bit changes from busy to not-busy.

ISP SPI Page Data Done Interrupt

If enabled, this event based interrupt becomes active when the ISP SPI Data Busy status bit changes from busy to not-busy.

ISP Interrupt Clearing

Interrupt clearing depends on the configured interrupt acknowledge mode.

For Clear-by-Write mode, the interrupts are cleared by writing a '1' to the corresponding interrupt status register bit(s).

For Clear-on-Read mode, the interrupts are automatically cleared by reading the interrupt status register.

The ISP SPI Page Data Done Interrupt should only be used for Page Program instructions. For other ISP SPI instructions (including Page Read), the ISP SPI Instruction Done Interrupt should be used.

10.2.2 Local Bus User Interrupt

Interrupt assertion and clearing is scope of the User FPGA design.

If enabled, a low level on the local bus interrupt line (LB_INT#) generates the interrupt while a high level takes the interrupt away. There is an on-board pull-up resistor for the local bus interrupt line.

Interrupt clearing depends on the configured interrupt acknowledge mode.

For Clear-by-Write mode, the interrupt is cleared when the User FPGA logic stops driving the LB_INT# line low or by disabling the interrupt.

For Clear-on-Read mode, the interrupt is automatically disabled by reading the interrupt status register.

10.2.3 Local Bus Error Interrupt

If enabled, the local bus error interrupt status is set when any of the following local bus error event flags is set:

- Local Bus PLL Loss-of-Lock Flag
- Local Bus Target Error Flag
- Local Bus Master Abort Flag

The Local Bus Interface Register may be read to determine which event flags caused the interrupt.

Interrupt clearing depends on the configured interrupt acknowledge mode.

For Clear-by-Write mode, the interrupt is cleared by writing a '1' to the corresponding interrupt status register bit.

For Clear-on-Read mode, the interrupt is automatically cleared by reading the interrupt status register.

11 LEDs

The TPMC634 provides three green LEDs on the cards back side.

LED Label	LED Description	Description
PG	Power Good Status LED	ON: +1.2V and +3.3V on-Board Power Supplies are indicating Power Good Status OFF: +1.2V and/or +3.3V on-Board Power Supplies are not indicating Power Good Status
XO	MachXO2 FPGA Status LED (PCI Target Chip)	This LED is SW controlled by a bit in the Control & Status Register By default (power-up & reset) the bit is set and the LED is On indicating MachXO2 PCI Target Chip operation
S6	Spartan-6 FPGA Status LED (User FPGA)	ON: Spartan-6 User FPGA is configured (operating) OFF: Spartan-6 User FPGA is not configured (not operating)

Table 11-1: On-Board LEDs

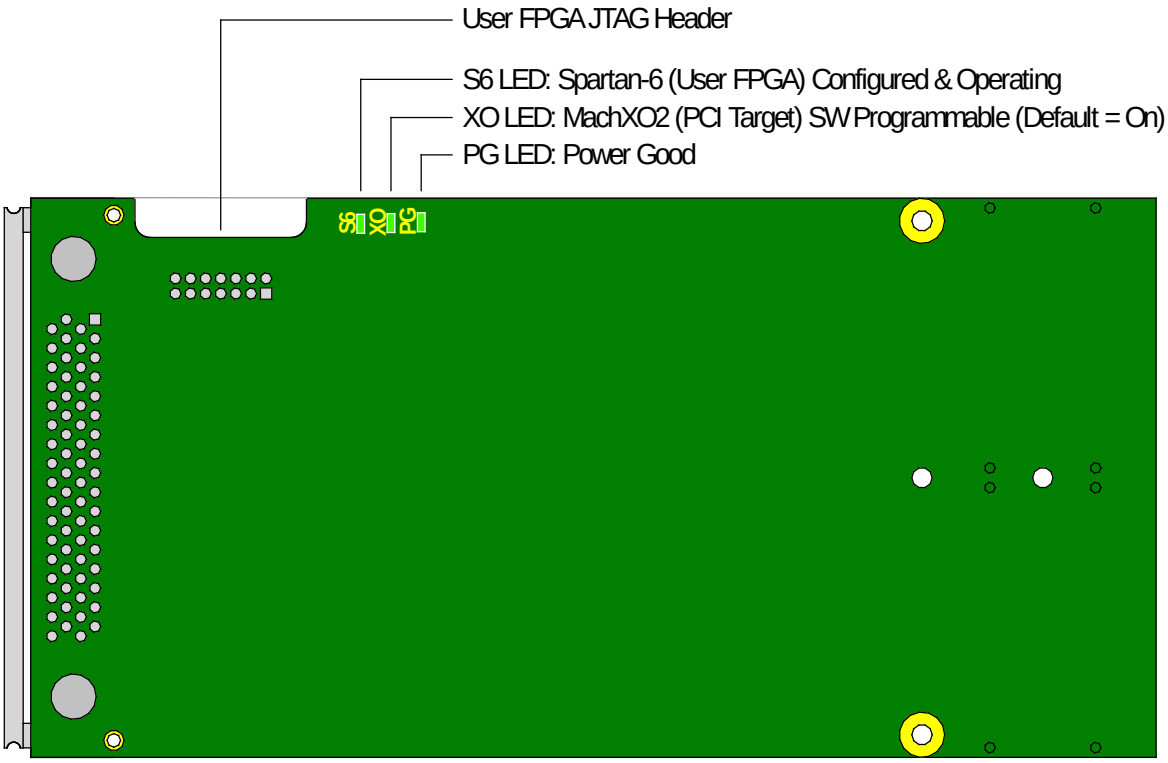


Figure 11-1: LED Location

12 JTAG Header

The TPMC634 provides a 14 pos., dual-row, 2mm pitch, right-angle JTAG header (near the 68 pos. front I/O connector) for accessing the Spartan-6 User FPGA JTAG port. The header mechanics and pin assignment support the direct connection of the Xilinx Platform Cable USB programmer 14 pos. 2mm flat ribbon cable.

The TPMC634 JTAG Header is intended to be used with the Xilinx iMPACT software and the Xilinx Platform Cable USB Programmer.

Pin-Count	14
Connector Type	2.00 mm Pitch Right Angle Header
Source & Order Info	Molex 877601416 or compatible

Table 12-1: JTAG Header Part Number

Pin	Signal	Direction	Remarks
1	NC		
2	+3.3V	From PMC	JTAG Reference Voltage
3	GND		
4	TMS	To PMC	3.3V Level expected
5	GND		
6	TCK	To PMC	3.3V Level expected
7	GND		
8	TDO	From PMC	3.3V Level
9	GND		
10	TDI	To PMC	3.3V Level expected
11	JTAG_CABLE_PRNST#	To PMC	This pin mates with a GND pin of the Xilinx Platform Cable USB Programmer
12	NC		
13	NC	To PMC	This pin mates with the PGND pin of the Xilinx Platform Cable USB II programmer
14	NC		

Table 12-2: JTAG Header Pin Assignment

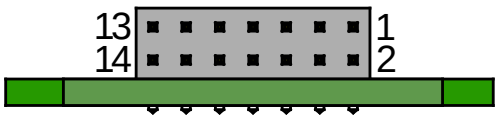


Figure 12-1: JTAG Header Pin Order

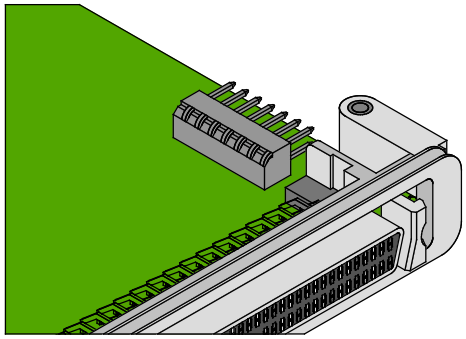


Figure 12-2: JTAG Header Location

13 Board HW-Configuration

13.1 Readable DIP-Switch

The TPMC634 provides a readable 4 position DIP-Switch.

The DIP-switch has no functional side-effects. The current DIP-Switch setting is readable in the DIP-Switch Register (Offset 0xF0) in the PCI Target Register Space (PCI BAR0) and may be used for identifying a certain TPMC634 card in a system.

The DIP-Switch default-setting is all positions OFF (reading value “0000”). For other settings, the DIP-Switch must be configured before installing the card on a PMC carrier.

13.2 TTL I/O Pull-Resistor Reference

Each TPMC634 TTL I/O line provides an on-board 4K7 pull resistor to a common configurable reference voltage.

The common pull resistor reference is configured by an on-board 4 position rotary switch.

The Rotary-Switch default-setting is ‘0’ (3.3V TTL I/O Pull Reference). For other settings, the Rotary-Switch must be configured before installing the card on a PMC carrier.

Rotary Switch Setting	TTL I/O Pull Resistor Reference
0	3.3V
1	5V
2	0V (GND)
3	Open

Table 13-1: 4-Pos. Rotary Switch Configuration

All TPMC634 I/O transceivers are disabled after power-up.

Each TPMC634 TTL I/O line provides an on-board pull-resistor to a common reference voltage. The common reference voltage is configurable by an on-board rotary switch.

The TPMC634 rotary switch configuration determines the default level of all TTL I/O lines after power-up! Factory default configuration is 3.3V.

14 I/O Interface

The TPMC634 I/O interface signals are available on both the 68 pin front-I/O connector and the 64 pin P14 rear-I/O connector.

Only one TPMC634 I/O interface must be used / installed at a time, either the front-I/O interface or the rear-I/O interface!

Do not use / install both the front I/O interface and the rear I/O interface at the same time!

14.1 General I/O Interface Description

The TPMC634 provides a total of 64 PMC I/O signals connected to User FPGA I/O pins via on-board I/O transceivers.

The same I/O signals are available at both the 68 pin front-I/O connector and at the 64 pin PMC P14 rear-I/O connector (for some of the P14 pins an assembly option provides using a ground connection instead of the I/O line signal connection). Only one TPMC634 I/O interface must be used / installed at a time (either the front I/O interface or the rear I/O interface).

For every PMC I/O line there are three corresponding I/O pins at the User FPGA. One FPGA I/O pin for the input data, one FPGA I/O pin for the output data and one FPGA I/O pin for output enable control of the on-Board output buffer/transmitter.

All PMC I/O lines are ESD protected (the ESD protection devices are located at the front I/O connector).

There are five order options which are differing regarding the I/O interface:

- The TPMC634-10R provides 64 single-ended TTL I/O lines
- The TPMC634-11R provides 32 differential RS485 I/O lines
- The TPMC634-12R provides a mix of 32 single-ended TTL I/O lines and 16 differential RS485 I/O lines
- The TPMC634-13R provides 32 differential M-LVDS I/O lines
- The TPMC634-14R provides a mix of 32 single-ended TTL I/O lines and 16 differential M-LVDS I/O lines

14.1.1 Single-Ended I/O Line Interface

For single-ended I/O lines (TPMC634-10R/-12R/-14R), 74LVT126 type buffers are used between the FPGA I/O and the PMC I/O interface. For every single-ended I/O line there is one 74LVT126 type buffer for the input signal and one 74LVT126 type buffer for the output signal. The 74LVT126 type buffer used for the input signal is always enabled. The 74LVT126 type buffer used for the output signal is enabled or disabled by a dedicated FPGA I/O signal. Each output enable control line has an on-board pull-down resistor to keep the output buffers disabled by default.

There is an on-board 47R series resistor in every single-ended I/O line. There also is an on-board 4K7 pull resistor on every single-ended I/O line. All the pull resistors share a common reference. The pull resistor reference is configurable by an on-board 4 position rotary switch. Pull resistor reference options are: 3.3V | 5V | GND | OPEN.

See the following figure for more information about the single-ended I/O circuit.

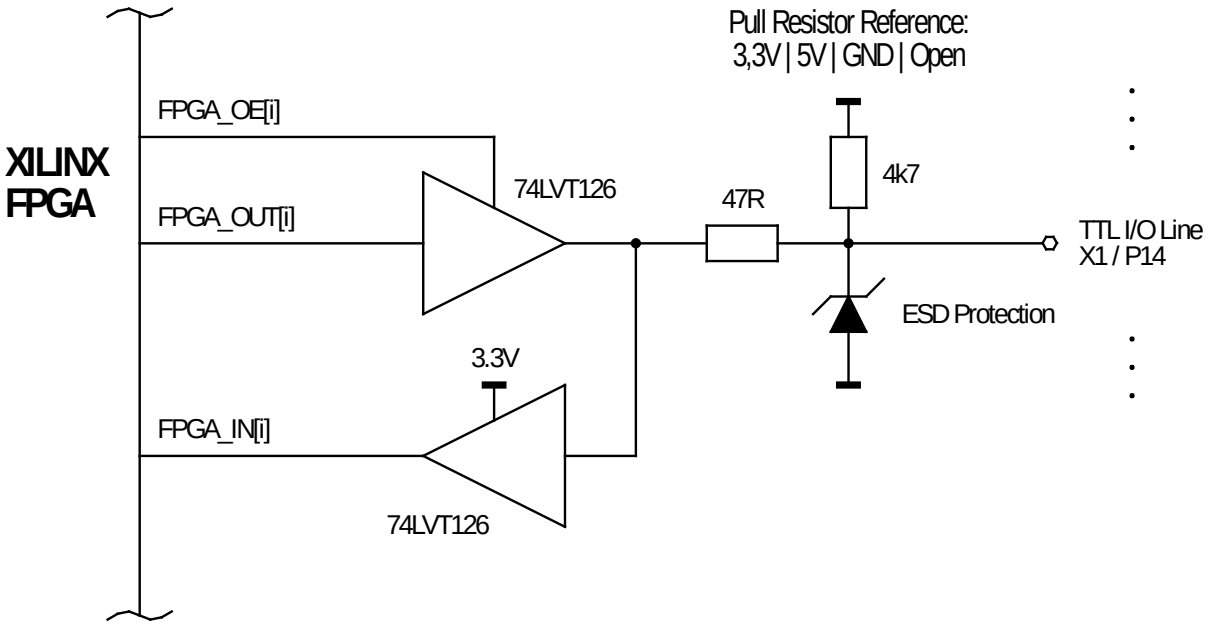


Figure 14-1: Single-Ended I/O Line Interface

14.1.1.1 Output Level & Output Current

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

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

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

All TPMC634 I/O transceivers are disabled after power-up.

Each TPMC634 TTL I/O line provides an on-board pull-resistor to a common reference voltage. The common reference voltage is configurable by an on-board rotary switch.

The TPMC634 rotary switch configuration determines the default level of all TTL I/O lines after power-up! Factory default configuration is 3.3V.

14.1.2 Differential I/O Line Interface

For differential I/O lines (TPMC634-11R/-12R/-13R/-14R) the on-board line transceiver type depends on the product variant.

On the TPMC634-11R and TPMC634-12R MAX3078E (or compatible) ESD protected RS485/RS422 transceivers are used as differential line transceivers.

On the TPMC634-13R and TPMC634-14R SN65MLVD206 (or compatible) M-LVDS Type 2 transceivers are used as differential line transceivers.

For every differential PMC I/O line there is an on-board termination resistor (no biasing).

See the following figure for more information about the differential I/O circuit.

Note that single-ended FPGA I/O pins are used for the differential PMC I/O interface.

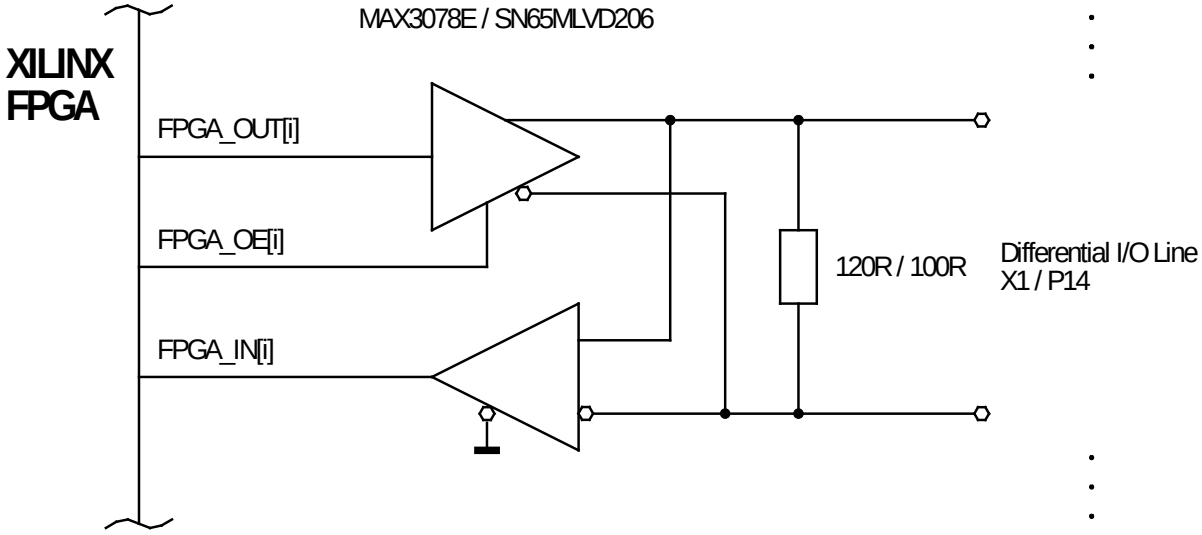


Figure 14-2: Differential I/O Line Interface

All TPMC634 I/O transceivers are disabled after power-up.

Each TPMC634 differential I/O line provides an on-board differential termination resistor.

14.2 I/O Connectors

14.2.1 Front-I/O Connector Part Number

Pin-Count	68
Connector Type	HD68 SCSI3
Source & Order Info	Tyco 787082-7 or compatible

Table 14-1 : Front I/O Connector Part Number

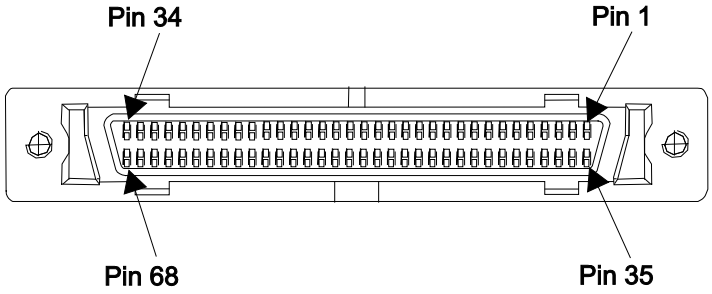


Figure 14-3: Front Panel I/O Connector Numbering

14.2.2 Rear-I/O Connector Part Number

Pin-Count	64
Connector Type	Mezzanine IEEE 1386 Plug
Source & Order Info	Molex: 787395-5 or compatible

Table 14-2: Rear I/O Connector Part Number

14.3 I/O Line Signal Level

TPMC634 Order Option	Line Signal Level
-10R	Single-Ended TTL
-11R	Differential RS485
-12R	Mixed Single-Ended TTL & Differential RS485
-13R	Differential M-LVDS
-14R	Mixed Single-Ended TTL & Differential M-LVDS

Table 14-3: I/O Line Signal Level

14.4 Front-I/O Pin Assignment

Pin	TPMC634 Order Option		
	-x0	-x1 & -x3	-x2 & -x4
1	IO_0	IO_0A/-	IO_0A/-
2	IO_2	IO_1A/-	IO_1A/-
3	IO_4	IO_2A/-	IO_2A/-
4	IO_6	IO_3A/-	IO_3A/-
5	IO_8	IO_4A/-	IO_4A/-
6	IO_10	IO_5A/-	IO_5A/-
7	IO_12	IO_6A/-	IO_6A/-
8	IO_14	IO_7A/-	IO_7A/-
9	GND	GND	GND
10	IO_16	IO_8A/-	IO_8A/-
11	IO_18	IO_9A/-	IO_9A/-
12	IO_20	IO_10A/-	IO_10A/-
13	IO_22	IO_11A/-	IO_11A/-
14	IO_24	IO_12A/-	IO_12A/-
15	IO_26	IO_13A/-	IO_13A/-
16	IO_28	IO_14A/-	IO_14A/-
17	IO_30	IO_15A/-	IO_15A/-
18	IO_32	IO_16A/-	IO_32
19	IO_34	IO_17A/-	IO_34
20	IO_36	IO_18A/-	IO_36
21	IO_38	IO_19A/-	IO_38
22	IO_40	IO_20A/-	IO_40
23	IO_42	IO_21A/-	IO_42
24	IO_44	IO_22A/-	IO_44
25	IO_46	IO_23A/-	IO_46
26	GND	GND	GND
27	IO_48	IO_24A/-	IO_48
28	IO_50	IO_25A/-	IO_50
29	IO_52	IO_26A/-	IO_52
30	IO_54	IO_27A/-	IO_54
31	IO_56	IO_28A/-	IO_56
32	IO_58	IO_29A/-	IO_58
33	IO_60	IO_30A/-	IO_60
34	IO_62	IO_31A/-	IO_62

Pin	TPMC634 Order Option		
	-x0	-x1 & -x3	-x2 & -x4
35	IO_1	IO_0B/+	IO_0B/+
36	IO_3	IO_1B/+	IO_1B/+
37	IO_5	IO_2B/+	IO_2B/+
38	IO_7	IO_3B/+	IO_3B/+
39	IO_9	IO_4B/+	IO_4B/+
40	IO_11	IO_5B/+	IO_5B/+
41	IO_13	IO_6B/+	IO_6B/+
42	IO_15	IO_7B/+	IO_7B/+
43	GND	GND	GND
44	IO_17	IO_8B/+	IO_8B/+
45	IO_19	IO_9B/+	IO_9B/+
46	IO_21	IO_10B/+	IO_10B/+
47	IO_23	IO_11B/+	IO_11B/+
48	IO_25	IO_12B/+	IO_12B/+
49	IO_27	IO_13B/+	IO_13B/+
50	IO_29	IO_14B/+	IO_14B/+
51	IO_31	IO_15B/+	IO_15B/+
52	IO_33	IO_16B/+	IO_33
53	IO_35	IO_17B/+	IO_35
54	IO_37	IO_18B/+	IO_37
55	IO_39	IO_19B/+	IO_39
56	IO_41	IO_20B/+	IO_41
57	IO_43	IO_21B/+	IO_43
58	IO_45	IO_22B/+	IO_45
59	IO_47	IO_23B/+	IO_47
60	GND	GND	GND
61	IO_49	IO_24B/+	IO_49
62	IO_51	IO_25B/+	IO_51
63	IO_53	IO_26B/+	IO_53
64	IO_55	IO_27B/+	IO_55
65	IO_57	IO_28B/+	IO_57
66	IO_59	IO_29B/+	IO_59
67	IO_61	IO_30B/+	IO_61
68	IO_63	IO_31B/+	IO_63

Table 14-4: TPMC634 Front I/O Pin Assignment

14.5 Rear-I/O Pin Assignment

Pin	TPMC634 Order Option		
	-x0	-x1 & -x3	-x2 & -x4
1	IO_0	IO_0A/-	IO_0A/-
3	IO_2	IO_1A/-	IO_1A/-
5	IO_4	IO_2A/-	IO_2A/-
7	IO_6	IO_3A/-	IO_3A/-
9	IO_8	IO_4A/-	IO_4A/-
11	IO_10	IO_5A/-	IO_5A/-
13	IO_12	IO_6A/-	IO_6A/-
15	IO_14	IO_7A/-	IO_7A/-
17	IO_16	IO_8A/-	IO_8A/-
19	IO_18	IO_9A/-	IO_9A/-
21	IO_20	IO_10A/-	IO_10A/-
23	IO_22	IO_11A/-	IO_11A/-
25	IO_24	IO_12A/-	IO_12A/-
27	IO_26	IO_13A/-	IO_13A/-
29	IO_28	IO_14A/-	IO_14A/-
31	IO_30	IO_15A/-	IO_15A/-
33	IO_32	IO_16A/-	IO_32
35	IO_34	IO_17A/-	IO_34
37	IO_36	IO_18A/-	IO_36
39	IO_38	IO_19A/-	IO_38
41	IO_40	IO_20A/-	IO_40
43	IO_42	IO_21A/-	IO_42
45	IO_44	IO_22A/-	IO_44
47	IO_46	IO_23A/-	IO_46
49	IO_48	IO_24A/-	IO_48
51	IO_50	IO_25A/-	IO_50
53	IO_52	IO_26A/-	IO_52
55	IO_54	IO_27A/-	IO_54
57	IO_56	IO_28A/-	IO_56
59	IO_58	IO_29A/-	IO_58
61	IO_60	IO_30A/-	IO_60
63	IO_62	IO_31A/-	IO_62

Pin	TPMC634 Order Option		
	-x0	-x1 & -x3	-x2 & -x4
2	IO_1	IO_0B/+	IO_0B/+
4	IO_3	IO_1B/+	IO_1B/+
6	IO_5	IO_2B/+	IO_2B/+
8	IO_7	IO_3B/+	IO_3B/+
10	IO_9	IO_4B/+	IO_4B/+
12	IO_11	IO_5B/+	IO_5B/+
14	IO_13	IO_6B/+	IO_6B/+
16	IO_15	IO_7B/+	IO_7B/+
18	IO_17	IO_8B/+	IO_8B/+
20	IO_19	IO_9B/+	IO_9B/+
22	IO_21	IO_10B/+	IO_10B/+
24	IO_23	IO_11B/+	IO_11B/+
26	IO_25	IO_12B/+	IO_12B/+
28	IO_27	IO_13B/+	IO_13B/+
30	IO_29	IO_14B/+	IO_14B/+
32	IO_31	IO_15B/+	IO_15B/+
34	IO_33	IO_16B/+	IO_33
36	IO_35	IO_17B/+	IO_35
38	IO_37	IO_18B/+	IO_37
40	IO_39	IO_19B/+	IO_39
42	IO_41	IO_20B/+	IO_41
44	IO_43	IO_21B/+	IO_43
46	IO_45	IO_22B/+	IO_45
48	IO_47	IO_23B/+	IO_47
50	IO_49	IO_24B/+	IO_49
52	IO_51	IO_25B/+	IO_51
54	IO_53	IO_26B/+	IO_53
56	IO_55	IO_27B/+	IO_55
58	IO_57	IO_28B/+	IO_57
60	IO_59	IO_29B/+	IO_59
62	IO_61	IO_30B/+	IO_61
64	IO_63	IO_31B/+	IO_63

I/O Pins 57 to 63 provide an assembly option for the GND signal instead of the I/O signal.

Table 14-5: TPMC634 Rear I/O Pin Assignment

15 Appendix A: User FPGA Port Map

The following may be used as the VHDL top level entity port map for the user FPGA application project.

```
entity tpmc634_xc6slx25_exa is
  port(

    -- Local Bus Interface Signals

    LB_RST_n   : in    std_logic;                     -- Reset
    LB_CLK     : in    std_logic;                     -- Clock
    LB_AD      : inout std_logic_vector(31 downto 0); -- Address/Data Bus
    LB_SEL_n   : in    std_logic_vector( 3 downto 0); -- Space Select
    LB_BE_n    : in    std_logic_vector( 3 downto 0); -- Byte Enables
    LB_CYC_n   : in    std_logic;                     -- Access Cycle
    LB_R_W_n   : in    std_logic;                     -- Direction
    LB_D_A_n   : in    std_logic;                     -- Address/Data Phase
    LB_MRDY_n  : in    std_logic;                     -- Master Ready
    LB_MABT_n  : in    std_logic;                     -- Master Abort
    LB_TRDY_n  : out   std_logic := '1';              -- Target Ready
    LB_TERR_n  : out   std_logic := '1';              -- Target Error
    LB_INT_n   : out   std_logic;                     -- Interrupt

    -- I/O Interface Signals

    IO_IN      : in    std_logic_vector(63 downto 0);
    IO_OUT     : out   std_logic_vector(63 downto 0) := (others => '0');
    IO_OE      : out   std_logic_vector(63 downto 0) := (others => '0');

    -- Other Signals

    BAUD_CLK   : in    std_logic;     -- 7.3728MHz Baud Rate Oscillator
    AUX_CLK    : in    std_logic      -- 33.25MHz +/-5% Auxiliary Clock

  );
end tpmc634_xc6slx25_exa;
```

16 Appendix B: User FPGA Constraint File

The following may be used as the Xilinx ISE design flow user constraint file (UCF) for the user FPGA application project.

```
##############################################################################
# User Constraints File for TPMC634 User FPGA Example
##############################################################################
#
# Project:       TPMC634 User FPGA Example
# Device:        XC6SLX25-2FGG484I
# Device Vendor: Xilinx
#
# Version:       1.0
# Date:          17.02.2015
#
##############################################################################
# General Constraints
##############################################################################
#
# Specifies the Vccaux voltage (Spartan-3A/6)
CONFIG VCCAUX = 3.3;
#
# Disable Suspend Mode (use AWAKE as User I/O Pin)
CONFIG ENABLE_SUSPEND = NO;
#
##############################################################################
# I/O Constraints
##############################################################################
#
# Specify the I/O Standard for a given port
NET "*" IOSTANDARD = LVCMOS33;
#
# Drive determines the drive strength of a device bi-directional or output pin.
NET "*" DRIVE = 8;
#
# Sets the SLOW output slew rate (rise/fall time) for an output
NET "*" SLEW = SLOW;
#
##############################################################################
# Location Constraints
##############################################################################
#
# Local Bus Interface
#
NET "LB_RST_n" LOC = AB17;
NET "LB_CLK" LOC = Y13; # PLL based Local Bus Clock (PCI Clock)
#NET "LB_CLK" LOC = Y12; # Oscillator based Local Bus Clock (Assembly Option)
CONFIG PROHIBIT = Y12; # Oscillator based Local Bus Clock (Assembly Option)
NET "LB_AD<0>" LOC = R11;
NET "LB_AD<1>" LOC = T11;
NET "LB_AD<2>" LOC = AA10;
NET "LB_AD<3>" LOC = AB10;
NET "LB_AD<4>" LOC = V11;
NET "LB_AD<5>" LOC = W11;
NET "LB_AD<6>" LOC = Y9;
```

```
NET "LB_AD<7>" LOC = AB9;
NET "LB_AD<8>" LOC = W10;
NET "LB_AD<9>" LOC = Y10;
NET "LB_AD<10>" LOC = AA8;
NET "LB_AD<11>" LOC = AB8;
NET "LB_AD<12>" LOC = W8;
NET "LB_AD<13>" LOC = V7;
NET "LB_AD<14>" LOC = W9;
NET "LB_AD<15>" LOC = Y8;
NET "LB_AD<16>" LOC = U9;
NET "LB_AD<17>" LOC = V9;
NET "LB_AD<18>" LOC = W6;
NET "LB_AD<19>" LOC = Y6;
NET "LB_AD<20>" LOC = Y5;
NET "LB_AD<21>" LOC = AB5;
NET "LB_AD<22>" LOC = AA4;
NET "LB_AD<23>" LOC = AB4;
NET "LB_AD<24>" LOC = Y3;
NET "LB_AD<25>" LOC = AB3;
NET "LB_AD<26>" LOC = R9;
NET "LB_AD<27>" LOC = R8;
NET "LB_AD<28>" LOC = T7;
NET "LB_AD<29>" LOC = R7;
NET "LB_AD<30>" LOC = U6;
NET "LB_AD<31>" LOC = V5;
NET "LB_SEL_n<0>" LOC = Y19;
NET "LB_SEL_n<1>" LOC = AB19;
NET "LB_SEL_n<2>" LOC = W18;
NET "LB_SEL_n<3>" LOC = Y18;
NET "LB_BE_n<0>" LOC = V15;
NET "LB_BE_n<1>" LOC = AA18;
NET "LB_BE_n<2>" LOC = AB18;
NET "LB_BE_n<3>" LOC = Y17;
NET "LB_CYC_n" LOC = AA21;
NET "LB_R_W_n" LOC = AA14;
NET "LB_D_A_n" LOC = AB14;
NET "LB_MRDY_n" LOC = Y16;
NET "LB_TRDY_n" LOC = V13;
NET "LB_MABT_n" LOC = W15;
NET "LB_TERR_n" LOC = W13;
NET "LB_INT_n" LOC = AB21;
#
# I/O Interface
#
NET "IO_IN<0>" LOC = Y2 | PULLDOWN;
NET "IO_IN<1>" LOC = W1 | PULLDOWN;
NET "IO_IN<2>" LOC = U4 | PULLDOWN;
NET "IO_IN<3>" LOC = M4 | PULLDOWN;
NET "IO_IN<4>" LOC = U3 | PULLDOWN;
NET "IO_IN<5>" LOC = T1 | PULLDOWN;
NET "IO_IN<6>" LOC = P2 | PULLDOWN;
NET "IO_IN<7>" LOC = N1 | PULLDOWN;
NET "IO_IN<8>" LOC = L3 | PULLDOWN;
NET "IO_IN<9>" LOC = K1 | PULLDOWN;
NET "IO_IN<10>" LOC = H4 | PULLDOWN;
NET "IO_IN<11>" LOC = H1 | PULLDOWN;
NET "IO_IN<12>" LOC = H6 | PULLDOWN;
NET "IO_IN<13>" LOC = F1 | PULLDOWN;
```

```
NET "IO_IN<14>" LOC = E3 | PULLDOWN;
NET "IO_IN<15>" LOC = D1 | PULLDOWN;
NET "IO_IN<16>" LOC = G6 | PULLDOWN;
NET "IO_IN<17>" LOC = H8 | PULLDOWN;
NET "IO_IN<18>" LOC = A2 | PULLDOWN;
NET "IO_IN<19>" LOC = J1 | PULLDOWN;
NET "IO_IN<20>" LOC = K5 | PULLDOWN;
NET "IO_IN<21>" LOC = J4 | PULLDOWN;
NET "IO_IN<22>" LOC = C5 | PULLDOWN;
NET "IO_IN<23>" LOC = C6 | PULLDOWN;
NET "IO_IN<24>" LOC = C7 | PULLDOWN;
NET "IO_IN<25>" LOC = A8 | PULLDOWN;
NET "IO_IN<26>" LOC = C9 | PULLDOWN;
NET "IO_IN<27>" LOC = D8 | PULLDOWN;
NET "IO_IN<28>" LOC = C13 | PULLDOWN;
NET "IO_IN<29>" LOC = D12 | PULLDOWN;
NET "IO_IN<30>" LOC = F13 | PULLDOWN;
NET "IO_IN<31>" LOC = G13 | PULLDOWN;
NET "IO_IN<32>" LOC = F14 | PULLDOWN;
NET "IO_IN<33>" LOC = C14 | PULLDOWN;
NET "IO_IN<34>" LOC = C15 | PULLDOWN;
NET "IO_IN<35>" LOC = C16 | PULLDOWN;
NET "IO_IN<36>" LOC = C17 | PULLDOWN;
NET "IO_IN<37>" LOC = A18 | PULLDOWN;
NET "IO_IN<38>" LOC = B10 | PULLDOWN;
NET "IO_IN<39>" LOC = A11 | PULLDOWN;
NET "IO_IN<40>" LOC = B12 | PULLDOWN;
NET "IO_IN<41>" LOC = C19 | PULLDOWN;
NET "IO_IN<42>" LOC = B22 | PULLDOWN;
NET "IO_IN<43>" LOC = D19 | PULLDOWN;
NET "IO_IN<44>" LOC = F19 | PULLDOWN;
NET "IO_IN<45>" LOC = C20 | PULLDOWN;
NET "IO_IN<46>" LOC = F20 | PULLDOWN;
NET "IO_IN<47>" LOC = E20 | PULLDOWN;
NET "IO_IN<48>" LOC = K17 | PULLDOWN;
NET "IO_IN<49>" LOC = H20 | PULLDOWN;
NET "IO_IN<50>" LOC = G20 | PULLDOWN;
NET "IO_IN<51>" LOC = K22 | PULLDOWN;
NET "IO_IN<52>" LOC = M21 | PULLDOWN;
NET "IO_IN<53>" LOC = N22 | PULLDOWN;
NET "IO_IN<54>" LOC = R20 | PULLDOWN;
NET "IO_IN<55>" LOC = T22 | PULLDOWN;
NET "IO_IN<56>" LOC = V21 | PULLDOWN;
NET "IO_IN<57>" LOC = N19 | PULLDOWN;
NET "IO_IN<58>" LOC = W20 | PULLDOWN;
NET "IO_IN<59>" LOC = K20 | PULLDOWN;
NET "IO_IN<60>" LOC = H22 | PULLDOWN;
NET "IO_IN<61>" LOC = J20 | PULLDOWN;
NET "IO_IN<62>" LOC = AA16 | PULLDOWN;
NET "IO_IN<63>" LOC = Y14 | PULLDOWN;
#
NET "IO_OUT<0>" LOC = Y1;
NET "IO_OUT<1>" LOC = T4;
NET "IO_OUT<2>" LOC = V3;
NET "IO_OUT<3>" LOC = V2;
NET "IO_OUT<4>" LOC = U1;
NET "IO_OUT<5>" LOC = R3;
NET "IO_OUT<6>" LOC = P1;
```

```
NET "IO_OUT<7>" LOC = M2;
NET "IO_OUT<8>" LOC = L1;
NET "IO_OUT<9>" LOC = K6;
NET "IO_OUT<10>" LOC = H3;
NET "IO_OUT<11>" LOC = G3;
NET "IO_OUT<12>" LOC = H5;
NET "IO_OUT<13>" LOC = G4;
NET "IO_OUT<14>" LOC = E1;
NET "IO_OUT<15>" LOC = C3;
NET "IO_OUT<16>" LOC = F5;
NET "IO_OUT<17>" LOC = B2;
NET "IO_OUT<18>" LOC = B3;
NET "IO_OUT<19>" LOC = M3;
NET "IO_OUT<20>" LOC = K4;
NET "IO_OUT<21>" LOC = A3;
NET "IO_OUT<22>" LOC = A5;
NET "IO_OUT<23>" LOC = B6;
NET "IO_OUT<24>" LOC = A7;
NET "IO_OUT<25>" LOC = D9;
NET "IO_OUT<26>" LOC = A9;
NET "IO_OUT<27>" LOC = D10;
NET "IO_OUT<28>" LOC = A13;
NET "IO_OUT<29>" LOC = H12;
NET "IO_OUT<30>" LOC = D13;
NET "IO_OUT<31>" LOC = E14;
NET "IO_OUT<32>" LOC = H14;
NET "IO_OUT<33>" LOC = B14;
NET "IO_OUT<34>" LOC = A15;
NET "IO_OUT<35>" LOC = B16;
NET "IO_OUT<36>" LOC = A17;
NET "IO_OUT<37>" LOC = E16;
NET "IO_OUT<38>" LOC = A10;
NET "IO_OUT<39>" LOC = D11;
NET "IO_OUT<40>" LOC = T19;
NET "IO_OUT<41>" LOC = B20;
NET "IO_OUT<42>" LOC = A20;
NET "IO_OUT<43>" LOC = D20;
NET "IO_OUT<44>" LOC = D21;
NET "IO_OUT<45>" LOC = C22;
NET "IO_OUT<46>" LOC = H19;
NET "IO_OUT<47>" LOC = E22;
NET "IO_OUT<48>" LOC = F21;
NET "IO_OUT<49>" LOC = T20;
NET "IO_OUT<50>" LOC = G22;
NET "IO_OUT<51>" LOC = L20;
NET "IO_OUT<52>" LOC = M22;
NET "IO_OUT<53>" LOC = P21;
NET "IO_OUT<54>" LOC = R22;
NET "IO_OUT<55>" LOC = U20;
NET "IO_OUT<56>" LOC = V22;
NET "IO_OUT<57>" LOC = P19;
NET "IO_OUT<58>" LOC = W22;
NET "IO_OUT<59>" LOC = K19;
NET "IO_OUT<60>" LOC = M20;
NET "IO_OUT<61>" LOC = K18;
NET "IO_OUT<62>" LOC = AB16;
NET "IO_OUT<63>" LOC = Y15;
#
```

```
NET "IO_OE<0>" LOC = W3;
NET "IO_OE<1>" LOC = T3;
NET "IO_OE<2>" LOC = M5;
NET "IO_OE<3>" LOC = V1;
NET "IO_OE<4>" LOC = T2;
NET "IO_OE<5>" LOC = R1;
NET "IO_OE<6>" LOC = N3;
NET "IO_OE<7>" LOC = M1;
NET "IO_OE<8>" LOC = K2;
NET "IO_OE<9>" LOC = J6;
NET "IO_OE<10>" LOC = H2;
NET "IO_OE<11>" LOC = G1;
NET "IO_OE<12>" LOC = F2;
NET "IO_OE<13>" LOC = F3;
NET "IO_OE<14>" LOC = D2;
NET "IO_OE<15>" LOC = C1;
NET "IO_OE<16>" LOC = J7;
NET "IO_OE<17>" LOC = B1;
NET "IO_OE<18>" LOC = J3;
NET "IO_OE<19>" LOC = L4;
NET "IO_OE<20>" LOC = K3;
NET "IO_OE<21>" LOC = A4;
NET "IO_OE<22>" LOC = D6;
NET "IO_OE<23>" LOC = A6;
NET "IO_OE<24>" LOC = B8;
NET "IO_OE<25>" LOC = C8;
NET "IO_OE<26>" LOC = D7;
NET "IO_OE<27>" LOC = C10;
NET "IO_OE<28>" LOC = E12;
NET "IO_OE<29>" LOC = F12;
NET "IO_OE<30>" LOC = H13;
NET "IO_OE<31>" LOC = F15;
NET "IO_OE<32>" LOC = D14;
NET "IO_OE<33>" LOC = A14;
NET "IO_OE<34>" LOC = D15;
NET "IO_OE<35>" LOC = A16;
NET "IO_OE<36>" LOC = B18;
NET "IO_OE<37>" LOC = D17;
NET "IO_OE<38>" LOC = C11;
NET "IO_OE<39>" LOC = C12;
NET "IO_OE<40>" LOC = A12;
NET "IO_OE<41>" LOC = B21;
NET "IO_OE<42>" LOC = A21;
NET "IO_OE<43>" LOC = F18;
NET "IO_OE<44>" LOC = D22;
NET "IO_OE<45>" LOC = G19;
NET "IO_OE<46>" LOC = H18;
NET "IO_OE<47>" LOC = J17;
NET "IO_OE<48>" LOC = F22;
NET "IO_OE<49>" LOC = J19;
NET "IO_OE<50>" LOC = K21;
NET "IO_OE<51>" LOC = L22;
NET "IO_OE<52>" LOC = N20;
NET "IO_OE<53>" LOC = P22;
NET "IO_OE<54>" LOC = T21;
NET "IO_OE<55>" LOC = U22;
NET "IO_OE<56>" LOC = M19;
NET "IO_OE<57>" LOC = P20;
```

```
NET "IO_OE<58>" LOC = L17;
NET "IO_OE<59>" LOC = H21;
NET "IO_OE<60>" LOC = L19;
NET "IO_OE<61>" LOC = J22;
NET "IO_OE<62>" LOC = W14;
NET "IO_OE<63>" LOC = AB15;
#
# Miscellaneous & Reserved
#
CONFIG PROHIBIT = AA2; # LB_REQ_n
CONFIG PROHIBIT = AB2; # LB_GNT_n
NET "BAUD_CLK" LOC = AB11; # 7.3728MHz Clock
NET "AUX_CLK" LOC = AB13; # AUX_CLK (33.25MHz +/-5%)
#
# Spartan-6 Configuration Port
#
CONFIG PROHIBIT = Y21;  # CCLK
CONFIG PROHIBIT = AA22; # M0_CMPMISO
CONFIG PROHIBIT = U15;  # M1
CONFIG PROHIBIT = Y7;   # D7
CONFIG PROHIBIT = Y4;   # D6
CONFIG PROHIBIT = W4;   # D5
CONFIG PROHIBIT = AB6;  # D4
CONFIG PROHIBIT = AA6;  # D3
CONFIG PROHIBIT = U13;  # D2_MISO3
CONFIG PROHIBIT = U14;  # D1_MISO2
CONFIG PROHIBIT = AA20; # D0_DIN_MISO_MISO1
CONFIG PROHIBIT = AB20; # MOSI_CSI_B_MISO0
CONFIG PROHIBIT = AB7;  # RDWR_B
CONFIG PROHIBIT = T6;   # INIT_B
CONFIG PROHIBIT = T5;   # CSO_B
#
# TPMC634 MachXO2 <> Spartan-6 Connections
#
CONFIG PROHIBIT = W12;  # MXO_S6_0
CONFIG PROHIBIT = Y11;  # MXO_S6_1
CONFIG PROHIBIT = AB12; # MXO_S6_2
CONFIG PROHIBIT = AA12; # MXO_S6_3
#
##############################################################################
# Timing Constraints
##############################################################################
#
# Local Bus Clock Cycle Timing (33 MHz)
NET "LB_CLK" TNM_NET = "LB_CLK";
TIMESPEC "TS_1" = PERIOD "LB_CLK" 30 ns HIGH 40%;
#
# Local Bus Input Setup Time
TIMEGRP "LOCAL_BUS_INPUTS" = PADS("LB_AD<*>" "LB_SEL_n<*>" "LB_BE_n<*>"
"LB_CYC_n" "LB_R_W_n" "LB_D_A_n" "LB_MRDY_n" "LB_MABT_n");
TIMEGRP "LOCAL_BUS_INPUTS" OFFSET = IN 10 ns VALID 11 ns BEFORE "LB_CLK" RISING;
#
# Local Bus Output Valid Time
#
TIMEGRP "LOCAL_BUS_AD_OUTPUTS" = PADS("LB_AD<*>");
TIMEGRP "LOCAL_BUS_AD_OUTPUTS" OFFSET = OUT 15 ns AFTER "LB_CLK" RISING;
#
TIMEGRP "LOCAL_BUS_CTRL_OUTPUTS" = PADS("LB_TRDY_n" "LB_TERR_n");
```

```
TIMEGRP "LOCAL_BUS_CTRL_OUTPUTS" OFFSET = OUT 12 ns AFTER "LB_CLK" RISING;
#
# NET "LB_RST_n" TIG;
#
# Auxiliary Clock (33.25 MHz)
NET "AUX_CLK" TNM_NET = "AUX_CLK";
TIMESPEC "TS_2" = PERIOD "AUX_CLK" 30 ns HIGH 40%;
#
# Baud Rate Clock (7.3728 MHz)
NET "BAUD_CLK" TNM_NET = "BAUD_CLK";
TIMESPEC "TS_3" = PERIOD "BAUD_CLK" 135 ns HIGH 40%;
#
```

17 Appendix C: Pre-Installed User FPGA Example Application

The TPMC634 Re-Configurable FPGA card comes with a pre-installed User FPGA Example application stored in the on-board SPI Flash.

After power-up the TPMC634 User FPGA automatically loads the FPGA configuration from the on-board SPI Flash and runs the TPMC634 User FPGA Example application.

The TPMC634 User FPGA Example application implements an accessible register set in User Space 0 (PCI BAR 2), mainly for controlling and reading the PMC I/O interface.

The source code for the TPMC634 User FPGA Example Application is available (as is) upon request (please send an email to support@tews.com).

Note that the TPMC634 Configuration EEPROM is factory pre-configured to enable the PCI BAR 2 (User Space 0) with a 256 KB size while the PCI BARs 3-5 (User Space 1-3) are disabled by default.

For User FPGA applications requiring only a single user space of up to 256 KB, there usually is no need to re-configure the TPMC634 Configuration EEPROM.

The factory pre-installed User FPGA Example application provides a 256 byte register space mapped multiple times into the pre-configured 256 KB User Space 0 (PCI BAR 2).

17.1 User FPGA Example Register Map

Offset to PCI BAR 2	Register Name
I/O Interface	
0x00	PMC I/O Input Register (Lower)
0x04	PMC I/O Input Register (Upper)
0x08	PMC I/O Output Register (Lower)
0x0C	PMC I/O Output Register (Upper)
0x10	PMC I/O Output Enable Register (Lower)
0x14	PMC I/O Output Enable Register (Upper)
0x18	Reserved
0x1C	Reserved
I/O Interrupt	
0x20	PMC I/O Rising Edge Interrupt Enable Register (Lower channels only)
0x24	PMC I/O Falling Edge Interrupt Enable Register (Lower channels only)
0x28	PMC I/O Interrupt Status Register (Lower channels only)
0x2C	Reserved
Reserved	
0x30 …	Reserved

0xDC	
Test	
0xE0	Clock Counter Control Register
0xE4	Clock Counter Register – Local Bus Clock
0xE8	Clock Counter Register – Auxiliary Clock
0xEC	Clock Counter Register – Baud Rate Clock
0xF0	Reserved
0xF4	Reserved
0xF8	General R/W Register
0xFC	Version Register

Table 17-1 : User FPGA Example Register Map

17.2 User FPGA Example Register Description

17.2.1 PMC I/O Input Register (Lower) (0x00)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_IN	PMC I/O Line [31:0] Input Register 0: Current PMC I/O Line State is Low 1: Current PMC I/O Line State is High The PMC I/O Line [31:0] inputs are capable of generating a PCI interrupt.	RO	-

Table 17-2: PMC I/O Input Register (Lower) (0x00)

17.2.2 PMC I/O Input Register (Upper) (0x04)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_IN	PMC I/O Line [63:32] Input Register 0: Current PMC I/O Line State is Low 1: Current PMC I/O Line State is High	RO	-

Table 17-3: PMC I/O Input Register (Upper) (0x04)

17.2.3 PMC I/O Output Register (Lower) (0x08)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_OUT	PMC I/O Line [31:0] Output Register 0: Set PMC I/O Line Output State Low 1: Set PMC I/O Line Output State High Note that for driving the configured output level, the corresponding line output must also be enabled.	R/W	all 0

Table 17-4: PMC I/O Output Register (Lower) (0x08)

17.2.4 PMC I/O Output Register (Upper) (0x0C)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_OUT	PMC I/O Line [63:32] Output Register 0: Set PMC I/O Line Output State Low 1: Set PMC I/O Line Output State High Note that for driving the configured output level, the corresponding line output must also be enabled.	R/W	all 0

Table 17-5: PMC I/O Output Register (Upper) (0x0C)

17.2.5 PMC I/O Output Enable Register (Lower) (0x10)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_OE	PMC I/O Line [31:0] Output Enable Register 0: Disable PMC I/O Line Driver 1: Enable PMC I/O Line Driver These bits are controlling the enable inputs of the on-board I/O line drivers.	R/W	all 0

Table 17-6: PMC I/O Output Enable Register (Lower) (0x10)

17.2.6 PMC I/O Output Enable Register (Upper) (0x14)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_OE	PMC I/O Line [63:32] Output Enable Register 0: Disable PMC I/O Line Driver 1: Enable PMC I/O Line Driver These bits are controlling the enable inputs of the on-board I/O line drivers.	R/W	all 0

Table 17-7: PMC I/O Output Enable Register (Upper) (0x14)

17.2.7 PMC I/O Rising Edge Interrupt Enable Register (0x20)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_R_ INT_EN	Rising Edge Interrupt Enable for PMC I/O Line [31:0] Inputs 0: Disable PMC I/O Line Rising Edge Interrupt 1: Enable PMC I/O Line Rising Edge Interrupt	R/W	all 0

Table 17-8: PMC I/O Rising Edge Interrupt Enable Register (0x20)

17.2.8 PMC I/O Falling Edge Interrupt Enable Register (0x24)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_F_ INT_EN	Falling Edge Interrupt Enable for PMC I/O Line [31:0] Inputs 0: Disable PMC I/O Line Falling Edge Interrupt 1: Enable PMC I/O Line Falling Edge Interrupt	R/W	all 0

Table 17-9: PMC I/O Falling Edge Interrupt Enable Register (0x24)

17.2.9 PMC I/O Interrupt Status Register (0x28)

Bit	Symbol	Description	Access	Reset Value
31:0	IO_INT_ STAT	Interrupt Status Register for PMC I/O Inputs [31:0] 0: No Interrupt for PMC I/O Line 1: PMC I/O Line Interrupt If any of the interrupt status bits is set, the local bus user interrupt line is set to the active level (low). If all interrupt status bits are clear, the local bus user interrupt line is set to the idle level (high). The interrupt status bits are cleared by writing a '1'. To generate a PCI INTA# interrupt, local bus user interrupts must also be enabled in the PCI Target Register space (PCI BAR 0).	R/C	all 0

Table 17-10: PMC I/O Interrupt Status Register (0x28)

17.2.10 Clock Counter Control Register (0xE0)

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved	-	0
1		Clock Counter Clear 0: Clock Counters Ready for Counting 1: Clock Counters in Clearing State Setting this bit clears the clock counter registers. The bit must be clear for clock counting mode.	R/W	0
0		Clock Counter Enable 0: Stop Clock Counters 1: Run Clock Counters Setting this bit sets all clock counters to counting mode. Clearing this bit stops all clock counters. The clock counters must be stopped prior reading the clock counter registers!	R/W	0

Table 17-11: Clock Counter Control Register (0xE0)

17.2.11 Clock Counter Register – Local Bus Clock (0xE4)

Bit	Symbol	Description	Access	Reset Value
31:0	CCR_LB	Local Bus Clock Counter Register Register for counting Local Bus Clock cycles. The Local Bus Clock frequency matches the PMC PCI clock frequency (typically 33MHz).	RO	all 0

Table 17-12: Clock Counter Register – Local Bus Clock (0xE4)

17.2.12 Clock Counter Register – Auxiliary Clock (0xE8)

Bit	Symbol	Description	Access	Reset Value
31:0	CCR_AUX	Auxiliary Clock Counter Register Register for counting Auxiliary Clock cycles. The expected Auxiliary Clock frequency is 33.25MHz ±5%.	RO	all 0

Table 17-13: Clock Counter Register – Auxiliary Clock (0xE8)

17.2.13 Clock Counter Register – Baud Rate Clock (0xEC)

Bit	Symbol	Description	Access	Reset Value
31:0	CCR_BAUD	Baud Rate Clock Counter Register Register for counting Baud Rate Clock cycles. The expected Baud Rate Clock frequency is 7.3728MHz.	RO	all 0

Table 17-14: Clock Counter Register – Baud Rate Clock (0xEC)

17.2.14 General R/W Test Register (0xF8)

Bit	Symbol	Description	Access	Reset Value
31:0	TEST_RW	32-bit General R/W Test Register	R/W	all 0

Table 17-15: General R/W Test Register (0xF8)

17.2.15 Version Register (0xFC)

Bit	Symbol	Description	Access	Reset Value
31:0	VER	User FPGA Example Version	RO	-

Table 17-16: Version Register (0xFC)