



The Embedded I/O Company **TECHNOLOGIES**

TPMC644

**Re-Configurable FPGA
with 64 TTL I/O / 32 Diff. I/O**

Version 1.0

User Manual

Issue 1.0.0

November 2025

TEWS Technologies GmbH

Eggerstedter Weg 14, 25421 Pinneberg, Germany

Phone: +49 (0) 4101 4058 0

info@tews.com www.tews.com

TPMC644-10R

Re-Configurable FPGA with 64 TTL (LVTTTL) I/O

TPMC644-11R

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

TPMC644-12R

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

TPMC644-13R

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

TPMC644-14R

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

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

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

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

©2025 by TEWS Technologies GmbH

All trademarks mentioned are property of their respective owners.

Document History

Issue	Description	Date
1.0.0	Initial Issue	November 2025

Table of Contents

TPMC644	1
Document History	3
Table of Contents	4
List of Figures	6
List of Tables	7
1 PRODUCT DESCRIPTION	9
2 TECHNICAL SPECIFICATION	11
3 HANDLING AND OPERATION INSTRUCTIONS	12
3.1 ESD Protection	12
3.2 Forced Air Cooling	12
3.3 Rear I/O Maximum Voltage Level	12
4 TERMS AND DEFINITIONS	13
4.1 Register Bit Access Types	13
4.2 Signal Direction Types	13
4.3 Style Conventions	13
5 ADDRESSING	14
5.1 PCI Configuration Space	14
5.1.1 PCI Device Identification	14
5.1.2 PCI Base Address Registers	14
5.2 PCI BAR0: PCI Interface Device Register Space	15
5.2.1 Register Map	15
5.3 PCI BAR1: In-System Programming Data Space	25
5.4 PCI BARs 2-5: User FPGA Spaces	25
6 CONFIGURATION EEPROM	26
7 USER FPGA	28
7.1 FPGA Part	28
7.2 I/O Bank Supply & Supported I/O Standards	28
7.3 Bank/Pin to Signal Mapping	28
7.4 Signal to Bank/Pin Mapping	34
7.5 Signal Description	40
7.5.1 Clock Sources	42
7.5.2 Front I/O Interface Signals	43
7.5.3 Rear I/O Interface Signals	48
7.6 User FPGA Design Hints	49
7.6.1 Single-Ended Front I/O	49
7.6.2 User EEPROM	49
8 USER FPGA CONFIGURATION	50
8.1 User FPGA Configuration Options	50
8.1.1 Configuration Data Files	50
8.2 FPGA Configuration from SPI Flash	51
8.2.1 Auto-Configuration at Power-Up	51
8.2.2 SW Controlled Re-Configuration	51
8.2.3 SPI Flash Preparation	52
8.2.4 SPI Configuration Parameters	52
8.3 FPGA Configuration via PCI/Software	53
8.3.1 Direct FPGA Programming	53
8.4 FPGA Configuration via JTAG Connector	54

9	SPI FLASH PROGRAMMING	55
9.1	SPI Flash Notes	55
9.1.1	SPI Flash Device Type	55
9.1.2	SPI Flash Programming Options	55
9.2	SPI Flash Programming via PCI/Software	55
9.2.1	SPI Flash Program Data	55
9.2.2	SPI Flash Instructions	56
9.2.3	Steps for SPI Flash Chip Erase Operation	56
9.2.4	Steps for SPI Flash Sector Erase Operation	57
9.2.5	Steps for SPI Flash Program Operation	58
9.2.6	Steps for SPI Flash Read Operation	59
9.3	SPI Flash Programming via JTAG Header	59
10	LOCAL BUS	60
10.1	Local Bus Interface Notes	60
10.2	Local Bus Cycle Description	60
10.2.1	Local Bus Master Abort (Local Bus Time-Out)	61
10.2.2	Local Bus Target Error	61
10.3	Local Bus Interface Signal Description	62
10.3.1	LB_TTERM# Notes (for Future Implementations)	64
10.3.2	Local Bus Signal States	65
10.4	Local Bus Interface Timing	66
10.5	Local Bus Signal Protocol Example Diagrams	67
11	CONFIGURATION HINTS	69
11.1	Jumper Configuration	69
12	THERMAL CONSIDERATIONS	70
13	I/O INTERFACE DESCRIPTION	71
13.1	Overview	71
13.2	LED Description	71
13.3	JTAG Connector	72
13.3.1	Connector Type	72
13.3.2	Pin Assignment	73
13.4	Front I/O Interface	73
13.4.1	Single-Ended I/O (LVTTTL)	74
13.4.2	Differential I/O (RS422/485, M-LVDS)	75
13.5	Front I/O Connector	75
13.5.1	Connector Type	75
13.5.2	Front I/O Connector Pin Assignment TPMC644-10R	76
13.5.3	Front I/O Connector Pin Assignment TPMC644-11R/-13R	77
13.5.4	Front I/O Connector Pin Assignment TPMC644-12R/-14R	78
13.6	Rear I/O Interface	79
13.7	Rear I/O Connector	80
13.7.1	Connector Type	80
13.7.2	Pin Assignment	80
14	APPENDIX A (TPMC644 BRD)	81
15	APPENDIX B (PORT MAP EXAMPLE)	82
16	APPENDIX C (DESIGN CONSTRAINTS EXAMPLE)	83

List of Figures

FIGURE 1-1 : TPMC644 BLOCK DIAGRAM	10
FIGURE 10-1 : SINGLE WRITE ACCESS	67
FIGURE 10-2 : SINGLE WRITE ACCESS (W/ WAIT STATE)	67
FIGURE 10-3 : SINGLE READ ACCESS.....	67
FIGURE 10-4 : SINGLE READ ACCESS (W/ WAIT STATE)	68
FIGURE 10-5 : READ WITH MASTER ABORT (OR TARGET ERROR)	68
FIGURE 11-1 : JUMPER LOCATION	69
FIGURE 12-1 : TPMC644-10R WITH HEATSINK	70
FIGURE 13-1 : TPMC644-10R BOARD OVERVIEW	71
FIGURE 13-2 : LED LOCATION	72
FIGURE 13-3 : JTAG CONNECTOR LOCATION.....	72
FIGURE 13-4 : JTAG CONNECTOR PIN-OUT	72
FIGURE 13-5 : SINGLE-ENDED FRONT I/O LINE CIRCUIT.....	74
FIGURE 13-6 : DIFFERENTIAL FRONT I/O LINE CIRCUIT	75
FIGURE 13-7 : FRONT I/O CONNECTOR PIN-OUT	75

List of Tables

TABLE 1-1 : ORDER OPTION OVERVIEW.....	9
TABLE 2-1 : TECHNICAL SPECIFICATION.....	11
TABLE 4-1 : REGISTER BIT ACCESS TYPES	13
TABLE 4-2 : SIGNAL DIRECTION TYPES	13
TABLE 5-1 : PCI DEVICE IDENTIFICATION.....	14
TABLE 5-2 : PCI BASE ADDRESS REGISTERS.....	14
TABLE 5-3 : PCI INTERFACE DEVICE REGISTER SPACE MAP	15
TABLE 5-4 : LOCAL BUS INTERFACE REGISTER.....	16
TABLE 5-5 : CONFIGURATION EEPROM REGISTER	17
TABLE 5-6 : INTERRUPT ENABLE REGISTER.....	18
TABLE 5-7 : INTERRUPT STATUS REGISTER.....	19
TABLE 5-8 : INTERRUPT CONFIGURATION REGISTER	20
TABLE 5-9 : USER FPGA CONFIGURATION CONTROL/STATUS REGISTER	21
TABLE 5-10 : USER FPGA CONFIGURATION DATA REGISTER (SLAVE SELECTMAP)	21
TABLE 5-11 : ISP CONTROL REGISTER (SPI).....	22
TABLE 5-12: ISP CONFIGURATION REGISTER (SPI).....	22
TABLE 5-13: ISP COMMAND REGISTER (SPI)	23
TABLE 5-14: ISP STATUS REGISTER (SPI)	23
TABLE 5-15: BOARD CONTROL & STATUS REGISTER	24
TABLE 5-16: MODULE SERIAL NUMBER REGISTER	24
TABLE 5-17: FIRMWARE IDENTIFICATION REGISTER	24
TABLE 5-18 : USER FPGA SPACE OVERVIEW (FACTORY DEFAULT SETTING)	25
TABLE 6-1 : CONFIGURATION EEPROM MAP	26
TABLE 6-2 : CONFIGURATION EEPROM TO PCI BAR PARAMETER MAPPING (MSW).....	27
TABLE 6-3 : CONFIGURATION EEPROM TO PCI BAR PARAMETER MAPPING (LSW).....	27
TABLE 6-4 : PCI BAR TO USER LOCAL SPACE MAPPING.....	27
TABLE 7-1 : USER FPGA I/O BANKS	28
TABLE 7-2 : USER FPGA I/O BANK/PIN TO SIGNAL MAPPING	34
TABLE 7-3 : USER FPGA I/O SIGNAL TO I/O BANK/PIN MAPPING.....	40
TABLE 7-4 : USER FPGA SIGNAL DESCRIPTION	42
TABLE 7-5 : FRONT I/O LINE TO USER FPGA I/O SIGNAL MAPPING	43
TABLE 7-6 : USER FPGA I/O SIGNAL TO FRONT I/O LINE MAPPING (I)	44
TABLE 7-7 : USER FPGA I/O SIGNAL TO FRONT I/O LINE MAPPING (II)	46
TABLE 7-8 : USER FPGA FRONT I/O LINE TABLE	47
TABLE 7-9 : REAR I/O SIGNAL DESCRIPTION	48
TABLE 8-1 : SPI CONFIGURATION PROPERTIES	52
TABLE 10-1 : LOCAL BUS INTERFACE SIGNAL DESCRIPTION	64
TABLE 10-2 : LOCAL BUS MASTER SIGNAL STATES (DATA PHASE).....	65
TABLE 10-3 : LOCAL BUS TARGET SIGNAL STATES (DATA PHASE)	65
TABLE 10-4 : LOCAL BUS SIGNAL STATES	65
TABLE 10-5 : LOCAL BUS INTERFACE TIMING.....	66
TABLE 11-1 : JUMPER CONFIGURATION.....	69

TABLE 13-1 : LED DESCRIPTION	71
TABLE 13-2 : JTAG HEADER.....	72
TABLE 13-3 : JTAG CONNECTOR PIN ASSIGNMENT	73
TABLE 13-4 : ORDER OPTION FRONT I/O OVERVIEW	73
TABLE 13-5 : FRONT I/O CONNECTOR TYPE	75
TABLE 13-6 : FRONT I/O CONNECTOR PIN ASSIGNMENT TPMC644-10R	76
TABLE 13-7 : FRONT I/O CONNECTOR PIN ASSIGNMENT TPMC644-11R/-13R	77
TABLE 13-8 : FRONT I/O CONNECTOR PIN ASSIGNMENT TPMC644-12R/-14R	78
TABLE 13-9 : REAR I/O CONNECTOR TYPE	80
TABLE 13-10 : REAR I/O CONNECTOR PIN ASSIGNMENT	80

1 Product Description

TPMC644 is a standard single-width 32 bit PMC module providing a user programmable FPGA and single-ended and/or differential digital I/O (depending on order option).

The TPMC644 utilizes a 32 bit 33 MHz PCI interface. A dedicated PCI Interface Device is used as a bridge between the PCI bus and an on-board 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 TPMC644 provides a JTAG header compatible with the AMD JTAG programming adapter, exclusively for the User FPGA.

After power-up, the User FPGA always attempts auto-configuration from an on-board non-volatile SPI Flash. The SPI Flash device is in-system programmable via the PCI bus (PCI Interface FPGA functionality) and/or programmable via the JTAG header (using the appropriate AMD programming hard- and software).

Options for volatile (direct) User FPGA configuration are via JTAG (JTAG header) or in-system direct FPGA programming via the PCI bus (PCI Interface Device functionality).

User FPGA in-system debugging is supported via the on-board JTAG header.

Order option / feature overview:

Feature	Order Option	TPMC644
User FPGA	All	AMD Spartan-7 XC7S50
Front I/O (via On-Board Transceivers)	-10	64x LVTTTL/TTL
	-11	32x RS422/RS485
	-12	16x RS422/RS485 + 32x LVTTTL/TTL
	-13	32x M-LVDS
	-14	16x M-LVDS + 32x LVTTTL/TTL
Rear I/O (Direct FPGA I/O pins)	All	Default 32x (+2) 3.3V LVTTTL/LVCMOS33 Configurable for up to 16x (+1) 2.5V LVDS or up to 32x (+2) LVCMOS25 instead

Table 1-1 : Order Option Overview

On-board line transceivers are used for the front I/O interface (for FPGA I/O protection, 5V tolerance, driver strength, RS422/485 level, M-LVDS level). The front I/O line direction (either input or output) is configurable individually per I/O line.

The front I/O lines are available on a 68 pin SCSI-3 type front-I/O connector. All front I/O lines are ESD protected.

The TPMC644 front I/O signal levels and connector pin assignment are compatible to the former TEWS Technologies TPMC632 and TPMC634 cards.

The TPMC644 also provides P14 rear I/O (directly/unprotected connected User FPGA I/O pins) with single-ended 3.3V LVTTTL/LVCMOS or differential 2.5V LVDS support.

The TPMC644 comes with a Board Reference Design (BRD) for the User FPGA, factory programmed to the on-board SPI Flash.

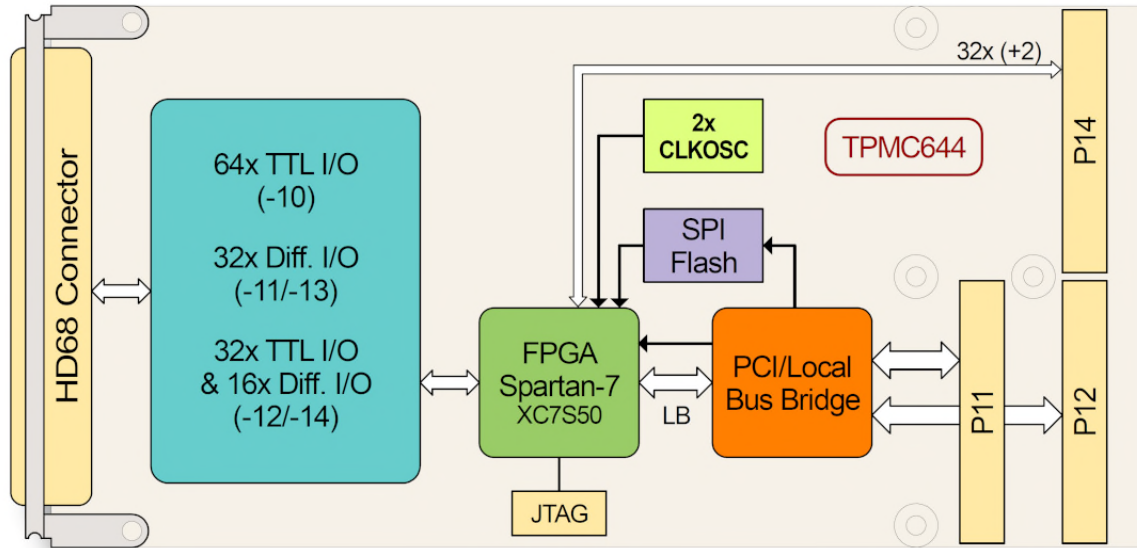


Figure 1-1 : TPMC644 Block Diagram

2 Technical Specification

General		
Mechanical Interface	PCI Mezzanine Card (PMC) conforming to IEEE P1386/P1386.1 Standard/Short Single-Width (149 mm x 74 mm)	
Electrical Interface	PCI Rev. 3.0 compatible 33 MHz / 32 bit PCI 3.3V and 5V PCI Signaling Voltage compatible	
Main On Board Devices		
PCI Interface Device	MachXO3 FPGA (Lattice)	
User FPGA	AMD Spartan-7 XC7S50	
Front I/O Transceiver	-10R	74LVC2G241
	-11R	THVD1450
	-12R	THVD1450, 74LVC2G241
	-13R	SN65MLVD206
	-14R	SN65MLVD206, 74LVC2G241
I/O Interface		
Front I/O Channels	-10R	64x 3.3V LVTTTL I/O, 5V tolerant, Common Pull-Option.
	-11R	32x RS485 I/O, On-Board 120R Line Termination.
	-12R	16x RS485 I/O, On-Board 120R Line Termination, 32x 3.3V LVTTTL I/O. 5V tolerant.
	-13R	32x M-LVDS I/O, On-Board 100R Line Termination.
	-14R	16x M-LVDS I/O, On-Board 100R Line Termination, 32x 3.3V LVTTTL I/O. 5V tolerant.
Rear I/O Channels	-xxR	32 + 2 Dedicated Direct FPGA I/O lines (unprotected)
Physical Data		
Power Requirements	<p>Lower Power Example, No external I/O load</p> <p>TPMC644-10R: 300 mA typical @ +5V DC (Front I/O: 64x driving on-board pull resistor)</p> <p>TPMC644-11R: 900 mA typical @ +5V DC (Front I/O: 32x driving on-board termination resistor)</p> <p>TPMC644-12R: 600 mA typical @ +5V DC (Front I/O driving 32x on-board pull resistor + 16x on-board termination resistor)</p> <p>Higher Power Example (User FPGA Design & I/O toggle, @33 MHz)</p> <p>TPMC644-11R: 1.85 A typical @ +5V DC</p>	
Temperature Range	Operating	-40°C to +85°C Adequate forced air cooling is required for ambient temperatures exceeding +25°C!
	Storage	-40°C to +125°C
MTBF	<p>TPMC644-10R: 441000 h</p> <p>TPMC644-11R: 496000 h</p> <p>TPMC644-12R: 465000 h</p> <p>MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G_B 20°C.</p> <p>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.</p>	
Humidity	5 – 95 % non-condensing	
Weight	115 g (including Heatsink)	

Table 2-1 : Technical Specification

3 Handling and Operation Instructions

3.1 ESD Protection



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

3.2 Forced Air Cooling



This PMC module requires adequate forced air cooling! Adequate forced air cooling is mandatory at ambient temperatures exceeding +25°C! Control/Verification of the User FPGA Junction Temperature for a given design and system environment is required! Without adequate forced air cooling, damage to the card can occur!

3.3 Rear I/O Maximum Voltage Level



CAUTION!
The rear I/O signals are directly connected to FPGA I/O pins without any additional on-board transceivers or protection. When used with a 3.3V bank supply (Default), applying external voltage levels > 3.6V to the rear I/O signals will damage the FPGA device!
When used with a 2.5V bank supply (Configurable by User FPGA Design and/or PCI Interface Device), applying external voltage levels > 2.625V to the rear I/O signals will damage the FPGA device!
In general, the card must be powered before or at the same time any externally generated signal level is applied to the rear I/O pins!

4 Terms and Definitions

4.1 Register Bit Access Types

Register Bit Access Type		Description
R	Read	The bit is readable by software (not writeable)
W	Write	The bit is writeable by software (not readable)
R/W	Read/Write	The bit is readable and writeable by software
R/C	Read/Clear	The bit is readable by software The bit is set by firmware Software may clear the bit by writing a '1'
R/S	Read/Set	The bit is readable by software Software may set this bit to '1' The bit is cleared by firmware

Table 4-1 : Register Bit Access Types

When reading reserved register bits, the read value is undefined.

For future software compatibility: For register write access, reserved bits shall be written '0'.

4.2 Signal Direction Types

Signal Direction Types as stated in Pin Assignment tables.

Signal Direction (Dir)	Description
I	Input Externally driven signal into the PMC
O	Output Signal driven out by PMC
I/O	Bi-Directional Signal
OD	Open Drain Output Signal driven low or tri-stated by PMC

Table 4-2 : Signal Direction Types

4.3 Style Conventions

Hexadecimal values are shown with prefix 0x (i.e. 0x029E).

Binary values are shown with prefix 0b (i.e. 0b0110).

"Active Low" signals are shown with a # suffix (i.e. RESET#).

5 Addressing

5.1 PCI Configuration Space

5.1.1 PCI Device Identification

The following table shows the factory default PCI ID values. The PCI ID values can be changed via the Configuration EEPROM.

ID	Offset	Factory Default Value
Vendor ID	0x00	0x1498 (TEWS Technologies)
Device ID	0x02	0x0284 (TPMC644)
Revision ID	0x08	0x00
Class Code	0x09	0x118000 (Other Data Acquisition/Signal Processing Controllers)
Subsystem Vendor ID	0x2C	0x1498 (TEWS Technologies)
Subsystem ID	0x2E	0x000A (TPMC644-10R) 0x000B (TPMC644-11R) 0x000C (TPMC644-12R) 0x000D (TPMC644-13R) 0x000E (TPMC644-14R)

Table 5-1 : PCI Device Identification

5.1.2 PCI Base Address Registers

PCI BAR	Offset in PCI Config Space	Space Mapping	Size (Byte)	Prefetch	Port Width (Bit)	Endian Mode	Description
0	0x10	MEM	512	No	32	Little	PCI Interface Device Register Space
1	0x14	MEM	256	No	32	Little	PCI Interface Device In-System-Programming Data Space
2	0x18	MEM	256K	No	32	Little	User FPGA Space 0 See BRD Documentation User (Re-) Configurable
3	0x1C	MEM	16K	No	32	Little	User FPGA Space 1 See BRD Documentation User (Re-) Configurable
4	0x20	Not used by factory default. May be used/configured by the user.					
5	0x24	Not used by factory default. May be used/configured by the user.					

BRD: User FPGA Board Reference Design

Table 5-2 : PCI Base Address Registers

PCI BARs 0 & 1 are fix by design and are physically located in the PCI Interface Device.

PCI BARs 2 to 3 (up to 5) are coming in a default factory setup and may be (re-) configured by the user via the Configuration EEPROM. Access to PCI BARs 2 to 3 (up to 5) is directed (by the PCI Interface Device) to the User FPGA device via the on-board local bus. These spaces are physically located in the User FPGA device.

5.2 PCI BAR0: PCI Interface Device Register Space

PCI BAR 0 is fix by design and is physically located in the PCI Interface Device.

5.2.1 Register Map

Offset to PCI BAR 0	Register Name	Size (Bit)
0x000 ... 0x08C	Reserved	32
0x090	Local Bus Interface Register	32
0x094 ... 0x09C	Reserved	32
0x0A0	I/O Control/Status Register	32
0x0A4 ... 0x0AC	Reserved	32
0x0B0	Configuration EEPROM Register	32
0x0B4 ... 0x0BC	Reserved	32
0x0C0	Interrupt Enable Register	32
0x0C4	Interrupt Status Register	32
0x0C8	Interrupt Configuration Register	32
0x0CC	Reserved	32
0x0D0	User FPGA Configuration Control/Status Register	32
0x0D4	User FPGA Configuration Data Register (Slave SelectMAP)	32
0x0D8 ... 0x0DC	Reserved	32
0x0E0	ISP Control Register	32
0x0E4	ISP Configuration Register (SPI)	32
0x0E8	ISP Command Register (SPI)	32
0x0EC	ISP Status Register (SPI)	32
0x0F0	Board Control/Status Register	32
0x0F4	Reserved	32
0x0F8	Card Serial Number Register	32
0x0FC	Firmware Version Register	32
Extended Registers (reserved on TPMC644)		
0x100 ... 0x1FC	Reserved	32

Table 5-3 : PCI Interface Device Register Space Map

5.2.1.1 Local Bus Interface Register (0x090)

Bit	Symbol	Description	Access	Reset Value
Event Flags				
31:28	-	Reserved	-	0
27	LB_TO	Local Bus Time-Out Flag Set by HW in case of a local bus access time-out (e.g. target does not respond for approx. 64 local bus clock cycles). Write '1' to clear the flag	R/C	0
26	LB_PLL_LOS	Local Bus PLL Loss-of-Lock Flag Set by HW when the local bus clock PLL lock status changes from locked to not-locked 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 Set by HW in case of a master initiated abort of a local bus cycle Write '1' to clear the flag	R/C	0
Status Bits				
23:18	-	Reserved	-	0
17	LB_PLL_ST	Local Bus Clock PLL Lock Status 0: PLL not locked 1: PLL locked	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 (or set into reset)	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	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 Disable state also clears latched event Note that in Disable state, a target not responding to a local bus may cause the PCI bus to hang!	R/W	0
0	LB_RST	Local Bus Reset 0: Do not force a SW controlled local bus reset 1: Force SW controlled local bus reset Only effects the Local Bus reset signal.	R/W	0

Table 5-4 : Local Bus Interface Register

5.2.1.2 I/O Control & Status Register (0x0A0)

This register is reserved.

5.2.1.3 Configuration EEPROM Register (0x0B0)

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/S	-
30	-	Reserved	-	0
29:28	EEP_OP	Configuration EEPROM Instruction-Code 10 → READ (a single word) 01 → WRITE (a single word) 11 → ERASE (a single word) 00 → WEN (EEP_A[9:8] = 11) WDS (EEP_A[9:8] = 00) ERAL (EEP_A[9:8] = 10) WRAL (EEP_A[9:8] = 01) WEN = Write Enable, WDS = Write Disable, ERAL = Erase All, WRAL = Write All	R/W	0
27:26	-	Reserved	-	0
25:16	EEP_A	Configuration EEPROM Address Address A[9:0] for x16 organized EEPROM Memory. 0x00 for first 16-bit EEPROM word, 0x01 for second 16-bit EEPROM word, ..., 0x3FF for last EEPROM word Required for READ and WRITE instructions. Max word address is 0x3FF (16kbit EEPROM → 2048 Bytes → 1024 Words). Address bits A[9:8] must be set accordingly for WEN WDS ERAL WRAL EEPROM instructions	R/W	0
15:0	EEP_D	Configuration EEPROM Read/Write Data Must be programmed with EEPROM write data prior to a WRITE instruction (may be set in the same register write that sets the EEP_GO_BSY bit). Presents the EEPROM read data after a READ instruction.	R/W	0

Table 5-5 : 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!

5.2.1.4 Interrupt Enable Register (0x0C0)

See *Interrupt Status Register* for more information.

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 (Masked only) 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 5-6 : Interrupt Enable Register

Disabling an interrupt automatically clears the corresponding interrupt status (except for the Local Bus User Interrupt).
Disabling the Local Bus User Interrupt only masks the interrupt. Clearing of the Local Bus User Interrupt condition is scope of the User FPGA design.

5.2.1.5 Interrupt Status Register (0x0C4)

Bit	Symbol	Description	Access	Reset Value
31:8	-	Reserved	-	0
Local Bus Interrupts				
7:6	-	Reserved	-	0
5	LB_ERR_IS	<p>Local Bus Error Interrupt Status</p> <p>0: Interrupt Status = Clear</p> <p>1: Interrupt Status = Set</p> <p>If enabled, the interrupt status is set when any of the following error events occurs:</p> <ul style="list-style-type: none"> • Local Bus PLL Loss-of-Lock Event • Local Bus Target Error Event • Local Bus Master Abort Event • Local Bus Timeout Event <p>The Local Bus Interface Register may be read to determine which event(s) caused the interrupt.</p> <p>See the <i>Interrupt Configuration Register</i> for interrupt clearing.</p>	R/C	0
4	LB_USR_IS	<p>Local Bus User Interrupt Status</p> <p>If enabled, the interrupt is active while the User FPGA drives the LB_INT# signal low.</p> <p>0: Interrupt Status = Clear (or Masked)</p> <p>1: Interrupt Status = Set</p> <p>This interrupt is only masked when being disabled. Interrupt source clearing is in scope of the User FPGA design.</p>	R	0
ISP Interrupts				
3:2	-	Reserved	-	0
1	ISP_INS_IS	<p>ISP SPI Instruction Done Event Interrupt Status</p> <p>0: Interrupt Status = Clear</p> <p>1: Interrupt Status = Set</p> <p>See the <i>Interrupt Configuration Register</i> for interrupt clearing.</p>	R/C	0
0	ISP_DAT_IS	<p>ISP SPI Page Data Done Event Interrupt Status</p> <p>0: Interrupt Status = Clear</p> <p>1: Interrupt Status = Set</p> <p>See the <i>Interrupt Configuration Register</i> for interrupt clearing.</p>	R/C	0

Table 5-7 : Interrupt Status Register

When any Interrupt Status Register bit is set, the PCI INTA# interrupt line is asserted.

5.2.1.6 Interrupt Configuration Register (0x0C8)

Bit	Symbol	Description	Access	Reset Value
31:1	-	Reserved	-	0
0	INT_ACK_MODE	<p>Interrupt Acknowledge Mode 0: Interrupt Clear-by-Write Mode 1: Interrupt Clear-on-Read Mode</p> <p>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 and is masked by disabling the interrupt.</p> <p>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 and the Local Bus User Interrupt is automatically disabled (masked).</p>	R/W	0

Table 5-8 : Interrupt Configuration Register

5.2.1.7 User FPGA Configuration Control/Status Register (0x0D0)

Bit	Symbol	Description	Access	Reset Value
31:4	-	Reserved	-	0
3	UFP_INIT_STAT	<p>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.</p>	R	x
2	UFP_DONE_STAT	<p>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 and low while the FPGA is not properly configured or unconfigured.</p>	R	x

1	UFP_RE_CFG	<p>User FPGA Re-Configuration Control</p> <p>0 → 1: Prepare a User FPGA Re-Configuration</p> <p>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 prepared as configured by UFP_CFG_MD.</p> <p>After setting this bit, SW should check the INIT_B and DONE signal status until both read as '0'.</p> <p>1 → 0: Start User FPGA Re-Configuration</p> <p>Effects: The User FPGA PROG_B signal is de-asserted again. After a short time the User FPGA releases the INIT_B pin, samples the configuration mode and starts the configuration sequence.</p> <p>After clearing this bit, SW should check the INIT_B signal status until it is read as '1'. DONE signal status '1' indicates successful User FPGA configuration.</p> <p>After power-up the FPGA automatically attempts to configure from the on-board SPI Flash in 'Master Serial / SPI' mode</p>	R/W	0
0	UFP_CFG_MD	<p>User FPGA Configuration Mode Control</p> <p>0: Master Serial / SPI (configuration from SPI Flash)</p> <p>1: Slave SelectMAP x8 (volatile configuration via PCI bus / software)</p> <p>Should be set accordingly before (and should not be changed when) UFP_RE_CFG is changed from 1 → 0</p> <p>The actual FPGA Re-Configuration is initiated by setting UFP_RE_CFG from 1 → 0</p>	R/W	0

Table 5-9 : 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!

5.2.1.8 User FPGA Configuration Data Register (Slave SelectMAP) (0x0D4)

Bit	Symbol	Description	Access	Reset Value
31:0	UFP_DAT	<p>Write Data Register for direct volatile User FPGA programming in x8 Slave SelectMAP mode (not for SPI Flash programming)</p> <p>Must be written with 32-bit User FPGA programming data until the User FPGA Done pin goes high</p> <p>Each register write loads four configuration data bytes to the User FPGA</p> <p>After writing the actual programming data, writing dummy data (0xFFFFFFFF) may be required to generate additional configuration clock cycles (until DONE is detected high).</p>	R/W	0

Table 5-10 : User FPGA Configuration Data Register (Slave SelectMAP)

The byte order send to the User FPGA Slave SelectMAP configuration data port is:

1. UFP_DAT[7:0], 2. UFP_DAT[15:8], 3. UFP_DAT[23:16], 4. UFP_DAT[31:24]

E.g. the AMD Sync Word 0xAA995566 must be mapped to the register byte lanes as follows:

0xAA → UFP_DAT[7:0], 0x99 → UFP_DAT[15:8], 0x55 → UFP_DAT[23:16], 0x66 → UFP_DAT[31:24].

5.2.1.9 ISP Control Register (0x0E0)

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 signal paths for the User FPGA configuration interface and the on-board SPI Flash. Must be set to 1 for SPI Flash programming (via PCI) and Direct User FPGA 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 5-11 : ISP Control Register (SPI)

5.2.1.10 ISP Configuration Register (SPI) (0x0E4)

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	R/W	0

Table 5-12: ISP Configuration Register (SPI)

5.2.1.11 ISP Command Register (SPI) (0x0E8)

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved	-	0
1	ISP_SPI_BRK_CMD	<p>ISP SPI Break Instruction Command Bit</p> <p>Writing a '1' logs a request to break/cancel the current SPI Instruction.</p> <p>Check the Instruction Busy Bit in the ISP Status Register for SPI Instruction busy/done status.</p> <p>Ignored (lost) while the Instruction Busy Bit is clear in the ISP Status Register.</p> <p>Always read as '0'.</p>	W	0
0	ISP_SPI_INS_CMD	<p>ISP SPI Start Instruction Command Bit</p> <p>Writing a '1' sets the SPI Instruction Busy Bit in the ISP Status Register and starts the configured SPI instruction.</p> <p>Ignored (lost) while the Instruction Busy Bit is set in the ISP Status Register.</p> <p>Always read as '0'.</p>	W	0

Table 5-13: ISP Command Register (SPI)

5.2.1.12 ISP Status Register (SPI) (0x0EC)

Bit	Symbol	Description	Access	Reset Value
31:2	-	Reserved	-	0
1	ISP_SPI_INS_BSY	<p>ISP SPI Instruction Busy Status</p> <p>Set & Cleared automatically by HW.</p> <p>Includes SPI Flash internal program/erase times.</p> <p>When clear again after being set, a new ISP SPI instruction may be started.</p> <p>Capable of generating an event based interrupt.</p> <p>0: No ISP SPI Instruction in Progress</p> <p>1: ISP SPI Instruction in Progress</p>	R	0
0	ISP_SPI_DAT_BSY	<p>ISP SPI Data Busy Status</p> <p>Set & Cleared automatically by HW.</p> <p>Does not include SPI Flash internal program/erase times.</p> <p>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).</p> <p>Capable of generating an event based interrupt.</p> <p>0: No ISP SPI Data Transfer in Progress</p> <p>1: ISP SPI Data Transfer in Progress</p>	R	0

Table 5-14: ISP Status Register (SPI)

5.2.1.13 Board Control & Status Register (0x0F0)

Bit	Symbol	Description	Access	Reset Value
31:20	-	Reserved Control Write as '0'	R/W	0
19	USR2_LED	User FPGA LED 2 Status 0: USR2 LED Off 1: USR2 LED On Only effective when bit 16 is set	R/W	0
18	USR1_LED	User FPGA LED 1 Status 0: USR1 LED Off 1: USR1 LED On Only effective when bit 16 is set	R/W	0
17	-	Reserved	-	-
16	USR_LED_CTRL	User FPGA LED Control 0: User FPGA (Default) 1: PCI Interface Device (via bits 18, 19)	R/W	0
15:12	VCC_B_SEL	Back-IO Bank Voltage Selection 0b1110: 3.3V 0b1101: 2.5V Others : User FPGA controlled (Default) See caution notice in the <i>Handling and Operation Instructions!</i>	R/W	0
11:8	-	Reserved Status	R	0
7	BOARD_PG	Board Power Good Indicates the status of the board power good signal. 0: Power Not Good 1: Power Good	R	0
6:5	-	Reserved Status	R	0
4:0	BLD_OPT	Build Option For internal use only.	R	-

Table 5-15: Board Control & Status Register

5.2.1.14 Module Serial Number Register (0x0F8)

Bit	Symbol	Description	Access	Reset Value
31:0	MOD_SN	Module Serial Number Example 9612345 (dec) → 92AC39 (hex) 31:24: 00, 23:16: 92, 15:08: AC, 07:00: 39	R	-

Table 5-16: Module Serial Number Register

5.2.1.15 Firmware Identification Register (0x0FC)

Bit	Symbol	Description	Access	Reset Value
31:0	FwID	Firmware ID	R	-

Table 5-17: Firmware Identification Register

5.3 PCI BAR1: In-System Programming Data Space

PCI BAR 1 is fix by design and is physically located in the PCI Interface Device.

The In-System Programming (ISP) Data Space is a 256 byte space used for passing data for in-system programming of the User FPGA SPI Configuration Flash (not for direct FPGA programming in Slave SelectMAP configuration mode).

The ISP Data Space size covers an SPI Flash Memory Page. The supported SPI Flash read and write instructions are page-based.

For ISP write/program instructions, the programming data must be written (zero-based) to the ISP Data Space before the page program instruction is started. The programming data must cover a complete SPI Flash memory page (256 Byte).

For ISP read instructions, the data can be read (zero-based) from the ISP Data Space after the page read instruction is done. The data is passed for a complete SPI Flash memory page (256 Byte).

Control, Configuration and Status registers for (non-volatile) In-System SPI Flash programming are located in the PCI Interface Device Register Space (BAR0). Control/Status and Data registers for direct (volatile) FPGA in-system programming in Slave SelectMAP configuration mode are also located in the PCI Interface Device Register Space (BAR0).

5.4 PCI BARs 2-5: User FPGA Spaces

The User FPGA Space resources are physically located within the User FPGA.

The TPMC644 supports up to four local spaces for mapping User FPGA address regions to the PCI bus (via the on-board local bus). Each local space is assigned to a dedicated PCI BAR.

PCI BARs 2 to 5 are available for implementing the user local spaces. Each user local space has a max size of up to 256MB and a dedicated space select coding during the address phase on the local bus interface.

PCI BARs 2 to 5 (User FPGA local spaces 0 to 3) are configurable (enable/disable, size) via the Board Configuration EEPROM, which is accessible via the PCI Interface Device Registers in PCI BAR0. When enabled, the PCI Interface Device forwards access to PCI BARs 2 to 5 to the User FPGA device via the on-board local bus

The actual User FPGA Space implementation depends on the User FPGA logic design and the corresponding Configuration EEPROM programming. The TPMC644 comes with a factory default configuration as shown below.

PCI BAR	Local Space	Local Space Select Coding L_AD[29:28]	Address Space Size		Description (Factory Default Configuration)
			Factory Default	User Max	
2	0	00	256 Kbyte	256 Mbyte	BRD User FPGA Space 0
3	1	01	16 Kbyte	256 Mbyte	BRD User FPGA Space 1
4	2	10	Not Used	256 Mbyte	Not used
5	3	11	Not Used	256 Mbyte	Not used

Table 5-18 : User FPGA Space Overview (Factory Default Setting)

See the TPMC644 BRD (Board Reference Design) documentation for more information about the factory default function for User FPGA Spaces 0 & 1.

6 Configuration EEPROM

The following configuration parameters are loaded from the on-board serial Configuration EEPROM at power-up and upon subsequent PCI resets.

- TPMC644 PCI Identifier (including Order Option)
- TPMC644 PCI BAR mapping of User FPGA Local Spaces (Enable, Size)

The EEPROM comes with a factory default setting. The user may (need to) adapt the EEPROM content according to the user application. The Board Configuration EEPROM is user programmable via a PCI BAR0 Register.

If any of the first two EEPROM words is read as 0xFFFF the EEPROM data is discarded (considered corrupted) and the listed parameters are set to fall-back values (invalid EEPROM column below). If both first EEPROM words differ from 0xFFFF the listed parameters are set according to the EEPROM content.

Serial EEPROM Word Offset	Parameter	Factory Default EEPROM Content	Invalid EEPROM Fall-Back Values
PCI Identifier			
0x000	PCI Device ID	0x0284	0x0284
0x001	PCI Vendor ID	0x1498	0x1498
0x002	PCI Class Code	0x1180	0x1180
0x003	PCI Class Code / Rev	0x0000	0x0000
0x004	PCI Subsystem ID	Order Option	0x0000
0x005	PCI Subsystem Vendor ID	0x1498	0x0000
0x006	Reserved	...	
0x007	Reserved	...	
PCI BAR Setup for User Spaces			
0x008	MSW PCI BAR 2 / Local User Space 0	0xFFFC	0x0000
0x009	LSW PCI BAR 2 / Local User Space 0	0x0000	0x0000
0x00A	MSW PCI BAR 3 / Local User Space 1	0xFFFF	0x0000
0x00B	LSW PCI BAR 3 / Local User Space 1	0xC000	0x0000
0x00C	MSW PCI BAR 4 / Local User Space 2	0x0000	0x0000
0x00D	LSW PCI BAR 4 / Local User Space 2	0x0000	0x0000
0x00E	MSW PCI BAR 5 / Local User Space 3	0x0000	0x0000
0x00F	LSW PCI BAR 5 / Local User Space 3	0x0000	0x0000
Reserved			
0x010 ... 0x3FF	Reserved		...

Table 6-1 : Configuration EEPROM Map

Note that for an invalid or blank EEPROM, the card is still identified as a TPMC644, but there is no valid card variant coding and there also is no PCI BAR enabled for any local user space.

The factory default EEPROM configuration is mapping a 256 Kbyte User FPGA space (intended as general register space) in PCI BAR2 and a 16Kbyte space in PCI BAR3. If this configuration would be sufficient for the User FPGA application, there typically would be no need for the user to modify the Configuration EEPROM values.

To disable a PCI BAR (and the corresponding PCI mapped User FPGA Local Space), the corresponding BAR configuration parameter must be set to 0x0000.0000 in the Configuration EEPROM.

To enable a PCI BAR BAR (and the corresponding PCI mapped User FPGA Local Space), the corresponding BAR configuration Parameter bits [31:28] must be set as “1111” and bits [2:1] must be set as “00”. If bits [31:28] or bits [2:1] are set otherwise, the BAR is disabled.

If enabled, PCI BAR bits [27:4, 3, 0] are taken from the corresponding EEPROM values, with bits [27:4] defining the space size.

EEPROM MSW PCI BAR	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	BAR Size											
PCI BAR Parameter [31:16]	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	1	1	1	1	BAR Size											

Table 6-2 : Configuration EEPROM to PCI BAR Parameter Mapping (MSW)

EEPROM LSW PCI BAR	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR Size												B3	0	0	B0
PCI BAR Parameter [15:0]	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	BAR Size												B3	0	0	B0

Table 6-3 : Configuration EEPROM to PCI BAR Parameter Mapping (LSW)

The BAR Size is configurable via 24 bits defined in the EEPROM MSW and LSW words. Leading ‘1’s are defining the address lines reserved for space select encoding and trailing ‘0’s are defining the address lines used for address decoding within the space.

If enabled, the actual BAR size is $2^{\text{number_of_trailing_zeros_in_BAR_Size}} \times 16$ Byte

Smallest BAR Size setting is 0b1111.1111.1111.1111.1111.1111 for 16 Byte.

Largest BAR Size setting is 0b0000.0000.0000.0000.0000.0000 for 256 Mbyte.

Example: A BAR Size setting of 0b1111.1111.0000.0000.0000.0000 corresponds to a BAR size of $2^{16} \times 16$ Byte = 1 Mbyte.

Bit 3 indicates a “Prefetchable” (data) space to the Host (i.e. a BAR with no side effects on reads). Typically bit 3 is set to 0 for a Register Space type of BAR.

Bit 0 is the Memory Space / IO Space indicator and should always be set to 0 (**strong recommendation**, PCI I/O Space is legacy).

The PCI BAR to User Local Space mapping/association is fix. PCI BARs shall be used/utilized in increasing order without omission.

PCI BAR	User Local Space
2	0
3	1
4	2
5	3

Table 6-4 : PCI BAR to User Local Space Mapping

7 User FPGA

7.1 FPGA Part

The TPMC644 User FPGA is a AMD Spartan-7 XC7S50 (Speedgrade -2).

The XC7S50 provides 52.160 logic cells, 8.150 slices (each slice providing 4 LUTs and 8 flip-flops), up to 600 Kb distributed RAM and up to 2.700 Kb Block RAM bits.

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

The XC7S50 provides 6 I/O banks which are used/utilized as follows.

Bank	(Main) Usage	Supply Voltage	Supported I/O Standards
0	Dedicated / Configuration	+3.3V	N/A for User I/O
14	Configuration Signals, Front I/O Data & Transceiver Control	+3.3V	LVTTL, LVCMOS33
15	Front I/O Transceiver Control, Rear I/O Data	+3.3V (Default)	LVTTL, LVCMOS33
		+2.5V (Configurable)	LVCMOS25, LVDS_25
16	Front I/O Data & Transceiver Control	+3.3V	LVTTL, LVCMOS33
34	Local Bus	+3.3V	LVTTL, LVCMOS33
35	Front I/O Data	+3.3V	LVTTL, LVCMOS33

Table 7-1 : User FPGA I/O Banks

All User FPGA I/O signals must use the LVTTL or LVCMOS33 I/O Standard.

The only exception are the Rear I/O signals. For the Rear I/O signals the supported I/O Standards depend on the application and configured I/O bank supply. See table above.

7.3 Bank/Pin to Signal Mapping

In the table below, signal direction (Dir) is from the User FPGA point of view. For outputs, the corresponding drive strength and slew rate settings are shown.

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
Bank 14						
14	N15	I/O	FIO_42	8	SLOW	
14	M21	-	Reserved			Do Not Use!
14	M22	-	Reserved			Do Not Use!
14	N21	-	Reserved			Do Not Use!
14	N22	-	Reserved			Do Not Use!
14	M19	-	Reserved			Do Not Use!
14	N19	I	EMC_CLK			
14	P21	-	Reserved			Do Not Use!
14	P22	-	Reserved			Do Not Use!
14	N20	-	Reserved			Do Not Use!
14	P20	-	Reserved			Do Not Use!

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
14	N17	-	Reserved			Do Not Use!
14	N18	I/O	FIO_43	8	SLOW	
14	T21	I/O	FIO_44	8	SLOW	
14	T22	I/O	FIO_45	8	SLOW	
14	V21	I/O	FIO_46	8	SLOW	
14	W22	I/O	FIO_47	8	SLOW	
14	U22	I/O	FIO_48	8	SLOW	
14	V22	I/O	FIO_49	8	SLOW	
14	W21	I/O	FIO_50	8	SLOW	
14	Y21	I/O	FIO_51	8	SLOW	
14	U20	I/O	FIO_52	8	SLOW	
14	V20	I/O	FIO_53	8	SLOW	
14	U18	I/O	FIO_54	8	SLOW	
14	U19	I/O	FIO_55	8	SLOW	
14	P15	I/O	SCL	8	SLOW	On-Board Pull-Up
14	P16	I/O	SDA	8	SLOW	On-Board Pull-Up
14	P17	O	ULED1	8	SLOW	
14	R18	O	ULED2	8	SLOW	
14	T17	-	Reserved			Do Not Use!
14	U17	-	Reserved			Do Not Use!
14	R16	-	Reserved			Do Not Use!
14	R17	-	Reserved			Do Not Use!
14	R19	I/O	FIO_63	8	SLOW	
14	R20	O	FIO_OE_63	8	SLOW	On-Board Pull-Down
14	T19	I/O	FIO_62	8	SLOW	
14	T20	O	FIO_OE_62	8	SLOW	On-Board Pull-Down
14	Y20	I/O	FIO_61	8	SLOW	
14	AA20	O	FIO_OE_61	8	SLOW	On-Board Pull-Down
14	Y22	I/O	FIO_60	8	SLOW	
14	AA22	O	FIO_OE_60	8	SLOW	On-Board Pull-Down
14	W18	I/O	FIO_59	8	SLOW	
14	Y19	O	FIO_OE_59	8	SLOW	On-Board Pull-Down
14	AA21	I/O	FIO_58	8	SLOW	
14	AB21	O	FIO_OE_58	8	SLOW	On-Board Pull-Down
14	AB19	I/O	FIO_57	8	SLOW	
14	AB20	I/O	FIO_56	8	SLOW	
14	V18	I	BLD_OPT2			
14	V19	I	BLD_OPT1			
14	T16	I	BLD_OPT0			
Bank 15						
15	A21	O	FIO_OE_42	8	SLOW	On-Board Pull-Down
15	F17	I/O	RIO_10/5+	8	SLOW	
15	F18	I/O	RIO_11/5-	8	SLOW	
15	H17	I/O	RIO_12/6+	8	SLOW	
15	G17	I/O	RIO_13/6-	8	SLOW	

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
15	J17	I/O	RIO_14/7+	8	SLOW	
15	H18	I/O	RIO_15/7-	8	SLOW	
15	H16	I/O	RIO_4/2+	8	SLOW	
15	G16	I/O	RIO_5/2-	8	SLOW	
15	K16	I/O	RIO_6/3+	8	SLOW	
15	J16	I/O	RIO_7/3-	8	SLOW	
15	K15	I/O	RIO_0/0+	8	SLOW	
15	J15	I/O	RIO_1/0-	8	SLOW	
15	B21	O	FIO_OE_43	8	SLOW	On-Board Pull-Down
15	B22	O	FIO_OE_44	8	SLOW	On-Board Pull-Down
15	D20	O	FIO_OE_45	8	SLOW	On-Board Pull-Down
15	C20	O	FIO_OE_46	8	SLOW	On-Board Pull-Down
15	D21	O	FIO_OE_47	8	SLOW	On-Board Pull-Down
15	C22	O	FIO_OE_48	8	SLOW	On-Board Pull-Down
15	E22	O	FIO_OE_49	8	SLOW	On-Board Pull-Down
15	D22	O	FIO_OE_50	8	SLOW	On-Board Pull-Down
15	F21	O	FIO_OE_51	8	SLOW	On-Board Pull-Down
15	F22	O	FIO_OE_52	8	SLOW	On-Board Pull-Down
15	F19	I/O	RIO_22/11+	8	SLOW	
15	F20	I/O	RIO_23/11-	8	SLOW	
15	M15	I/O	RIO_2/1+	8	SLOW	
15	L15	I/O	RIO_3/1-	8	SLOW	
15	M17	I/O	RIO_16/8+	8	SLOW	
15	M18	I/O	RIO_17/8-	8	SLOW	
15	L16	I/O	RIO_8/4+	8	SLOW	
15	K17	I/O	RIO_9/4-	8	SLOW	
15	K18	I/O	RIO_18/9+	8	SLOW	
15	K19	I/O	RIO_19/9-	8	SLOW	
15	J19	I/O	RIO_24/12+	8	SLOW	
15	J20	I/O	RIO_25/12-	8	SLOW	
15	L18	I/O	RIO_20/10+	8	SLOW	
15	L19	I/O	RIO_21/10-	8	SLOW	
15	H19	I/O	RIO_26/13+	8	SLOW	
15	G20	I/O	RIO_27/13-	8	SLOW	
15	G21	I/O	RIO_62/31+	8	SLOW	
15	G22	I/O	RIO_63/31-	8	SLOW	
15	H20	I/O	RIO_28/14+	8	SLOW	
15	H21	I/O	RIO_29/14-	8	SLOW	
15	L22	O	FIO_57_OE	8	SLOW	On-Board Pull-Down
15	K22	O	FIO_56_OE	8	SLOW	On-Board Pull-Down
15	J21	O	FIO_55_OE	8	SLOW	On-Board Pull-Down
15	J22	O	FIO_54_OE	8	SLOW	On-Board Pull-Down
15	L20	I/O	RIO_30/15+	8	SLOW	
15	L21	I/O	RIO_31/15-	8	SLOW	
15	M16	O	FIO_OE_53	8	SLOW	On-Board Pull-Down

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
Bank 16						
16	G13	I/O	FIO_17	8	SLOW	
16	G11	O	FIO_OE_17	8	SLOW	On-Board Pull-Down
16	F11	I/O	FIO_18	8	SLOW	
16	G10	O	FIO_OE_18	8	SLOW	On-Board Pull-Down
16	F10	I/O	FIO_19	8	SLOW	
16	E11	O	FIO_OE_19	8	SLOW	On-Board Pull-Down
16	D11	I/O	FIO_20	8	SLOW	
16	F13	O	FIO_OE_20	8	SLOW	On-Board Pull-Down
16	F14	I/O	FIO_21	8	SLOW	
16	F12	O	FIO_OE_21	8	SLOW	On-Board Pull-Down
16	E12	I/O	FIO_22	8	SLOW	
16	G14	O	FIO_OE_22	8	SLOW	On-Board Pull-Down
16	F15	I/O	FIO_23	8	SLOW	
16	D13	O	FIO_OE_23	8	SLOW	On-Board Pull-Down
16	C13	I/O	FIO_24	8	SLOW	
16	A11	O	FIO_OE_24	8	SLOW	On-Board Pull-Down
16	A12	I/O	FIO_25	8	SLOW	
16	B13	O	FIO_OE_25	8	SLOW	On-Board Pull-Down
16	A13	I/O	FIO_26	8	SLOW	
16	C10	O	FIO_OE_26	8	SLOW	On-Board Pull-Down
16	C11	I/O	FIO_27	8	SLOW	
16	D14	O	FIO_OE_27	8	SLOW	On-Board Pull-Down
16	D15	I/O	FIO_28	8	SLOW	
16	D12	O	FIO_OE_28	8	SLOW	On-Board Pull-Down
16	C12	I/O	FIO_29	8	SLOW	
16	C15	O	FIO_OE_29	8	SLOW	On-Board Pull-Down
16	C16	I/O	FIO_30	8	SLOW	
16	B15	O	FIO_OE_30	8	SLOW	On-Board Pull-Down
16	B16	I/O	FIO_31	8	SLOW	
16	D16	O	FIO_OE_31	8	SLOW	On-Board Pull-Down
16	C17	I/O	FIO_32	8	SLOW	
16	E15	O	FIO_OE_32	8	SLOW	On-Board Pull-Down
16	E16	I/O	FIO_33	8	SLOW	
16	B14	O	FIO_OE_33	8	SLOW	On-Board Pull-Down
16	A14	I/O	FIO_34	8	SLOW	
16	A16	O	FIO_OE_34	8	SLOW	On-Board Pull-Down
16	A17	I/O	FIO_35	8	SLOW	
16	D18	O	FIO_OE_35	8	SLOW	On-Board Pull-Down
16	C18	I/O	FIO_36	8	SLOW	
16	C19	O	FIO_OE_36	8	SLOW	On-Board Pull-Down
16	B19	I/O	FIO_37	8	SLOW	
16	A18	O	FIO_OE_37	8	SLOW	On-Board Pull-Down
16	A19	I/O	FIO_38	8	SLOW	
16	E17	O	FIO_OE_38	8	SLOW	On-Board Pull-Down

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
16	E18	I/O	FIO_39	8	SLOW	
16	B20	O	FIO_OE_39	8	SLOW	On-Board Pull-Down
16	A20	I/O	FIO_40	8	SLOW	
16	E19	O	FIO_OE_40	8	SLOW	On-Board Pull-Down
16	D19	I/O	FIO_41	8	SLOW	
16	G15	O	FIO_OE_41	8	SLOW	On-Board Pull-Down
Bank 34						
34	R2	I/O	LB_AD0	12	SLOW	
34	W5	I	LB_RST#			
34	W4	I	LB_CYC#			
34	T7	I	LB_R/W#			
34	T6	I	LB_D/A#			
34	V5	I	LB_C/BE#0			
34	V4	I	LB_C/BE#1			
34	T5	I	LB_C/BE#2			
34	U5	I	LB_C/BE#3			
34	V7	I	LB_MRDY#			
34	V6	I	LB_MLAST#			
34	T8	I	LB_MABT#			
34	U8		Reserved			Do Not Use!
34	V1	I/O	LB_AD1	12	SLOW	
34	W1	I/O	LB_AD2	12	SLOW	
34	W2	I/O	LB_AD3	12	SLOW	
34	Y1	I/O	LB_AD4	12	SLOW	
34	U2	I/O	LB_AD5	12	SLOW	
34	U1	I/O	LB_AD6	12	SLOW	
34	W3	O	LB_TRDY#	8	SLOW	
34	Y3	O	LB_TTERM#	8	SLOW	
34	U4	O	LB_TERR#	8	SLOW	
34	U3	O, OD	LB_INT#	8	SLOW	On-Board Pull-Up
34	T3	I	LB_CLK	8	SLOW	
34	T2	I/O	LB_AD7	12	SLOW	
34	AA6	I/O	LB_AD8	12	SLOW	
34	AB6	I/O	LB_AD9	12	SLOW	
34	Y6	-	Reserved			Do Not Use!
34	Y5	-	Reserved			Do Not Use!
34	Y4	-	Reserved			Do Not Use!
34	AA3	I/O	LB_AD10	12	SLOW	
34	AB3	I/O	LB_AD11	12	SLOW	
34	AB2	I/O	LB_AD12	12	SLOW	
34	AA2	I/O	LB_AD13	12	SLOW	
34	AA1	I/O	LB_AD14	12	SLOW	
34	AB5	I/O	LB_AD15	12	SLOW	
34	AB4	I/O	LB_AD16	12	SLOW	
34	V8	-	Reserved			Do Not Use!

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
34	W8	-	Reserved			Do Not Use!
34	Y8	-	Reserved			Do Not Use!
34	AA8	I/O	LB_AD17	12	SLOW	
34	W7	-	Reserved			Do Not Use!
34	Y7	-	Reserved			Do Not Use!
34	AA10	-	Reserved			Do Not Use!
34	AB10	I/O	LB_AD18	12	SLOW	
34	AA7	I/O	LB_AD19	12	SLOW	
34	AB7	I/O	LB_AD20	12	SLOW	
34	AA9	I/O	LB_AD21	12	SLOW	
34	AB9	I/O	LB_AD22	12	SLOW	
34	T1	I/O	LB_AD23	12	SLOW	
Bank 35						
35	H7	O, OD	RIO_VCCO_SEL			On-Board Pull-Up
35	J3	-	Reserved			Do Not Use!
35	H2	-	Reserved			Do Not Use!
35	H4	-	Reserved			Do Not Use!
35	H3	-	Reserved			Do Not Use!
35	J2	I/O	LB_AD31	12	SLOW	
35	J1	I/O	LB_AD30	12	SLOW	
35	K3	I/O	FIO_0	8	SLOW	
35	K2	O	FIO_OE_0	8	SLOW	On-Board Pull-Down
35	H6	I/O	FIO_1	8	SLOW	
35	H5	O	FIO_OE_1	8	SLOW	On-Board Pull-Down
35	K6	I/O	FIO_2	8	SLOW	
35	J6	O	FIO_OE_2	8	SLOW	On-Board Pull-Down
35	J8	I/O	FIO_3	8	SLOW	
35	J7	O	FIO_OE_3	8	SLOW	On-Board Pull-Down
35	L7	I/O	FIO_4	8	SLOW	
35	L6	O	FIO_OE_4	8	SLOW	On-Board Pull-Down
35	L5	I/O	FIO_5	8	SLOW	
35	K5	O	FIO_OE_5	8	SLOW	On-Board Pull-Down
35	L4	I/O	FIO_6	8	SLOW	
35	K4	O	FIO_OE_6	8	SLOW	On-Board Pull-Down
35	L8	I/O	FIO_7	8	SLOW	
35	K8	O	FIO_OE_7	8	SLOW	On-Board Pull-Down
35	M8	I	USR_CLK			40 MHz
35	M7	-	Reserved			Do Not Use!
35	M4	I	BRG_CLK			7.3728 MHz
35	M3	I/O	FIO_8	8	SLOW	
35	L1	I/O	LB_AD29	12	SLOW	
35	K1	I/O	LB_AD28	12	SLOW	
35	M2	I/O	LB_AD27	12	SLOW	
35	M1	I/O	LB_AD26	12	SLOW	
35	N4	O	FIO_OE_8	8	SLOW	On-Board Pull-Down

Bank	Pin	Dir	Signal	Drive [mA]	Slew Rate	Comment
35	N3	I/O	FIO_9	8	SLOW	
35	P1	I/O	LB_AD25	12	SLOW	
35	N1	I/O	LB_AD24	12	SLOW	
35	P3	O	FIO_OE_9	8	SLOW	On-Board Pull-Down
35	P2	I/O	FIO_10	8	SLOW	
35	P8	O	FIO_OE_10	8	SLOW	On-Board Pull-Down
35	N8	I/O	FIO_11	8	SLOW	
35	N5	O	FIO_OE_11	8	SLOW	On-Board Pull-Down
35	M5	I/O	FIO_12	8	SLOW	
35	P6	O	FIO_OE_12	8	SLOW	On-Board Pull-Down
35	N6	I/O	FIO_13	8	SLOW	
35	P7	O	FIO_OE_13	8	SLOW	On-Board Pull-Down
35	N7	I/O	FIO_14	8	SLOW	
35	R7	O	FIO_OE_14	8	SLOW	On-Board Pull-Down
35	R6	I/O	FIO_15	8	SLOW	
35	R5	O	FIO_OE_15	8	SLOW	On-Board Pull-Down
35	R4	I/O	FIO_16	8	SLOW	
35	R3	O	FIO_OE_16	8	SLOW	On-Board Pull-Down

Table 7-2 : User FPGA I/O Bank/Pin to Signal Mapping

7.4 Signal to Bank/Pin Mapping

In the table below, signal direction (Dir) is from the User FPGA point of view. For outputs, the corresponding drive strength and slew rate settings are shown.

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
Local Bus Signals (Address/Data Bus)						
LB_AD0	34	R2	I/O	12	SLOW	
LB_AD1	34	V1	I/O	12	SLOW	
LB_AD2	34	W1	I/O	12	SLOW	
LB_AD3	34	W2	I/O	12	SLOW	
LB_AD4	34	Y1	I/O	12	SLOW	
LB_AD5	34	U2	I/O	12	SLOW	
LB_AD6	34	U1	I/O	12	SLOW	
LB_AD7	34	T2	I/O	12	SLOW	
LB_AD8	34	AA6	I/O	12	SLOW	
LB_AD9	34	AB6	I/O	12	SLOW	
LB_AD10	34	AA3	I/O	12	SLOW	
LB_AD11	34	AB3	I/O	12	SLOW	
LB_AD12	34	AB2	I/O	12	SLOW	
LB_AD13	34	AA2	I/O	12	SLOW	
LB_AD14	34	AA1	I/O	12	SLOW	
LB_AD15	34	AB5	I/O	12	SLOW	
LB_AD16	34	AB4	I/O	12	SLOW	
LB_AD17	34	AA6	I/O	12	SLOW	

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
LB_AD18	34	AB10	I/O	12	SLOW	
LB_AD19	34	AA7	I/O	12	SLOW	
LB_AD20	34	AB7	I/O	12	SLOW	
LB_AD21	34	AA9	I/O	12	SLOW	
LB_AD22	34	AB9	I/O	12	SLOW	
LB_AD23	34	T1	I/O	12	SLOW	
LB_AD24	35	N1	I/O	12	SLOW	
LB_AD25	35	P1	I/O	12	SLOW	
LB_AD26	35	M1	I/O	12	SLOW	
LB_AD27	35	M2	I/O	12	SLOW	
LB_AD28	35	K1	I/O	12	SLOW	
LB_AD29	35	L1	I/O	12	SLOW	
LB_AD30	35	J1	I/O	12	SLOW	
LB_AD31	35	J2	I/O	12	SLOW	
Local Bus Signals (Control)						
LB_RST#	34	W5	I			
LB_CYC#	34	W4	I			
LB_R/W#	34	T7	I			
LB_D/A#	34	T6	I			
LB_C/BE#0	34	V5	I			
LB_C/BE#1	34	V4	I			
LB_C/BE#2	34	T5	I			
LB_C/BE#3	34	U5	I			
LB_MRDY#	34	V7	I			
LB_MLAST#	34	V6	I			
LB_MABT#	34	T8	I			
LB_TRDY#	34	W3	O	8	SLOW	
LB_TTERM#	34	Y3	O	8	SLOW	
LB_TERR#	34	U4	O	8	SLOW	
LB_INT#	34	U3	O, OD	8	SLOW	On-Board Pull-Up
LB_CLK	34	T3	I			
Misc. Signals						
BRG_CLK	35	M4	I			
USR_CLK	35	M8	I			
RIO_VCCO_SEL	35	H7	O	8	SLOW	On-Board Pull-Up
ULED1	14	P17	O	8	SLOW	
ULED2	14	R18	O	8	SLOW	
BLD_OPT2	14	V18	I			
BLD_OPT1	14	V19	I			
BLD_OPT0	14	T16	I			
SCL	14	P15	I/O	8	SLOW	On-Board Pull-Up
SDA	14	P16	I/O	8	SLOW	On-Board Pull-Up
Front I/O Signals (Data)						
FIO_0	35	K3	I/O	8	SLOW	

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
FIO_1	35	H6	I/O	8	SLOW	
FIO_2	35	K6	I/O	8	SLOW	
FIO_3	35	J8	I/O	8	SLOW	
FIO_4	35	L7	I/O	8	SLOW	
FIO_5	35	L5	I/O	8	SLOW	
FIO_6	35	L4	I/O	8	SLOW	
FIO_7	35	L8	I/O	8	SLOW	
FIO_8	35	M3	I/O	8	SLOW	
FIO_9	35	N3	I/O	8	SLOW	
FIO_10	35	P2	I/O	8	SLOW	
FIO_11	35	N8	I/O	8	SLOW	
FIO_12	35	M5	I/O	8	SLOW	
FIO_13	35	N6	I/O	8	SLOW	
FIO_14	35	N7	I/O	8	SLOW	
FIO_15	35	R8	I/O	8	SLOW	
FIO_16	35	R4	I/O	8	SLOW	
FIO_17	16	G13	I/O	8	SLOW	
FIO_18	16	F11	I/O	8	SLOW	
FIO_19	16	F10	I/O	8	SLOW	
FIO_20	16	D11	I/O	8	SLOW	
FIO_21	16	F14	I/O	8	SLOW	
FIO_22	16	E12	I/O	8	SLOW	
FIO_23	16	F15	I/O	8	SLOW	
FIO_24	16	C13	I/O	8	SLOW	
FIO_25	16	A12	I/O	8	SLOW	
FIO_26	16	A13	I/O	8	SLOW	
FIO_27	16	C11	I/O	8	SLOW	
FIO_28	16	D15	I/O	8	SLOW	
FIO_29	16	C12	I/O	8	SLOW	
FIO_30	16	C16	I/O	8	SLOW	
FIO_31	16	B16	I/O	8	SLOW	
FIO_32	16	C17	I/O	8	SLOW	
FIO_33	16	E16	I/O	8	SLOW	
FIO_34	16	A14	I/O	8	SLOW	
FIO_35	16	A17	I/O	8	SLOW	
FIO_36	16	C18	I/O	8	SLOW	
FIO_37	16	B19	I/O	8	SLOW	
FIO_38	16	A19	I/O	8	SLOW	
FIO_39	16	E18	I/O	8	SLOW	
FIO_40	16	A20	I/O	8	SLOW	
FIO_41	16	D19	I/O	8	SLOW	
FIO_42	14	N15	I/O	8	SLOW	
FIO_43	14	N18	I/O	8	SLOW	
FIO_44	14	T21	I/O	8	SLOW	
FIO_45	14	T22	I/O	8	SLOW	

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
FIO_46	14	V21	I/O	8	SLOW	
FIO_47	14	W22	I/O	8	SLOW	
FIO_48	14	U22	I/O	8	SLOW	
FIO_49	14	V22	I/O	8	SLOW	
FIO_50	14	W21	I/O	8	SLOW	
FIO_51	14	Y21	I/O	8	SLOW	
FIO_52	14	U20	I/O	8	SLOW	
FIO_53	14	V20	I/O	8	SLOW	
FIO_54	14	U18	I/O	8	SLOW	
FIO_55	14	U19	I/O	8	SLOW	
FIO_56	14	AB20	I/O	8	SLOW	
FIO_57	14	AB19	I/O	8	SLOW	
FIO_58	14	AA21	I/O	8	SLOW	
FIO_59	14	W18	I/O	8	SLOW	
FIO_60	14	Y22	I/O	8	SLOW	
FIO_61	14	Y20	I/O	8	SLOW	
FIO_62	14	T19	I/O	8	SLOW	
FIO_63	14	R19	I/O	8	SLOW	
Front I/O Signals (Output Enable aka Direction Control)						
FIO_OE_0	35	K2	O	8	SLOW	On-Board Pull-Down
FIO_OE_1	35	H5	O	8	SLOW	On-Board Pull-Down
FIO_OE_2	35	J6	O	8	SLOW	On-Board Pull-Down
FIO_OE_3	35	J7	O	8	SLOW	On-Board Pull-Down
FIO_OE_4	35	L6	O	8	SLOW	On-Board Pull-Down
FIO_OE_5	35	K5	O	8	SLOW	On-Board Pull-Down
FIO_OE_6	35	K4	O	8	SLOW	On-Board Pull-Down
FIO_OE_7	35	K8	O	8	SLOW	On-Board Pull-Down
FIO_OE_8	35	N4	O	8	SLOW	On-Board Pull-Down
FIO_OE_9	35	P3	O	8	SLOW	On-Board Pull-Down
FIO_OE_10	35	P8	O	8	SLOW	On-Board Pull-Down
FIO_OE_11	35	N5	O	8	SLOW	On-Board Pull-Down
FIO_OE_12	35	P6	O	8	SLOW	On-Board Pull-Down
FIO_OE_13	35	P7	O	8	SLOW	On-Board Pull-Down
FIO_OE_14	35	R7	O	8	SLOW	On-Board Pull-Down
FIO_OE_15	35	R5	O	8	SLOW	On-Board Pull-Down
FIO_OE_16	35	R3	O	8	SLOW	On-Board Pull-Down
FIO_OE_17	16	G11	O	8	SLOW	On-Board Pull-Down
FIO_OE_18	16	G10	O	8	SLOW	On-Board Pull-Down
FIO_OE_19	16	E11	O	8	SLOW	On-Board Pull-Down
FIO_OE_20	16	F13	O	8	SLOW	On-Board Pull-Down
FIO_OE_21	16	F12	O	8	SLOW	On-Board Pull-Down
FIO_OE_22	16	G14	O	8	SLOW	On-Board Pull-Down
FIO_OE_23	16	D13	O	8	SLOW	On-Board Pull-Down
FIO_OE_24	16	A11	O	8	SLOW	On-Board Pull-Down
FIO_OE_25	16	B13	O	8	SLOW	On-Board Pull-Down

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
FIO_OE_26	16	C10	O	8	SLOW	On-Board Pull-Down
FIO_OE_27	16	D14	O	8	SLOW	On-Board Pull-Down
FIO_OE_28	16	D12	O	8	SLOW	On-Board Pull-Down
FIO_OE_29	16	C15	O	8	SLOW	On-Board Pull-Down
FIO_OE_33	16	B15	O	8	SLOW	On-Board Pull-Down
FIO_OE_31	16	D16	O	8	SLOW	On-Board Pull-Down
FIO_OE_32	16	E15	O	8	SLOW	On-Board Pull-Down
FIO_OE_33	16	B14	O	8	SLOW	On-Board Pull-Down
FIO_OE_34	16	A16	O	8	SLOW	On-Board Pull-Down
FIO_OE_35	16	D18	O	8	SLOW	On-Board Pull-Down
FIO_OE_36	16	C19	O	8	SLOW	On-Board Pull-Down
FIO_OE_37	16	A18	O	8	SLOW	On-Board Pull-Down
FIO_OE_38	16	E17	O	8	SLOW	On-Board Pull-Down
FIO_OE_39	16	B20	O	8	SLOW	On-Board Pull-Down
FIO_OE_40	16	E19	O	8	SLOW	On-Board Pull-Down
FIO_OE_41	16	G15	O	8	SLOW	On-Board Pull-Down
FIO_OE_42	15	A21	O	8	SLOW	On-Board Pull-Down
FIO_OE_43	15	B21	O	8	SLOW	On-Board Pull-Down
FIO_OE_44	15	B22	O	8	SLOW	On-Board Pull-Down
FIO_OE_45	15	D20	O	8	SLOW	On-Board Pull-Down
FIO_OE_46	15	C20	O	8	SLOW	On-Board Pull-Down
FIO_OE_47	15	D21	O	8	SLOW	On-Board Pull-Down
FIO_OE_48	15	C22	O	8	SLOW	On-Board Pull-Down
FIO_OE_49	15	E22	O	8	SLOW	On-Board Pull-Down
FIO_OE_50	15	D22	O	8	SLOW	On-Board Pull-Down
FIO_OE_51	15	F21	O	8	SLOW	On-Board Pull-Down
FIO_OE_52	15	F22	O	8	SLOW	On-Board Pull-Down
FIO_OE_53	15	M16	O	8	SLOW	On-Board Pull-Down
FIO_OE_54	15	J22	O	8	SLOW	On-Board Pull-Down
FIO_OE_55	15	J21	O	8	SLOW	On-Board Pull-Down
FIO_OE_56	15	K22	O	8	SLOW	On-Board Pull-Down
FIO_OE_57	15	L22	O	8	SLOW	On-Board Pull-Down
FIO_OE_58	14	AB21	O	8	SLOW	On-Board Pull-Down
FIO_OE_59	14	Y19	O	8	SLOW	On-Board Pull-Down
FIO_OE_60	14	AA22	O	8	SLOW	On-Board Pull-Down
FIO_OE_61	14	AA20	O	8	SLOW	On-Board Pull-Down
FIO_OE_62	14	T20	O	8	SLOW	On-Board Pull-Down
FIO_OE_63	14	R20	O	8	SLOW	On-Board Pull-Down
Rear I/O Signals						
RIO_0/0+	15	K15	I/O	8	SLOW	
RIO_1/0-	15	J15	I/O	8	SLOW	
RIO_2/1+	15	M15	I/O	8	SLOW	
RIO_3/1-	15	L15	I/O	8	SLOW	
RIO_4/2+	15	H16	I/O	8	SLOW	
RIO_5/2-	15	G16	I/O	8	SLOW	

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
RIO_6/3+	15	K16	I/O	8	SLOW	
RIO_7/3-	15	J16	I/O	8	SLOW	
RIO_8/4+	15	L16	I/O	8	SLOW	
RIO_9/4-	15	K17	I/O	8	SLOW	
RIO_10/5+	15	F17	I/O	8	SLOW	
RIO_11/5-	15	F18	I/O	8	SLOW	
RIO_12/6+	15	H17	I/O	8	SLOW	
RIO_13/6-	15	G17	I/O	8	SLOW	
RIO_14/7+	15	J17	I/O	8	SLOW	
RIO_15/7-	15	H18	I/O	8	SLOW	
RIO_16/8+	15	M17	I/O	8	SLOW	
RIO_17/8-	15	M18	I/O	8	SLOW	
RIO_18/9+	15	K18	I/O	8	SLOW	
RIO_19/9-	15	K19	I/O	8	SLOW	
RIO_20/10+	15	L18	I/O	8	SLOW	
RIO_21/10-	15	L19	I/O	8	SLOW	
RIO_22/11+	15	F19	I/O	8	SLOW	
RIO_23/11-	15	F20	I/O	8	SLOW	
RIO_24/12+	15	J19	I/O	8	SLOW	
RIO_25/12-	15	J20	I/O	8	SLOW	
RIO_26/13+	15	H19	I/O	8	SLOW	
RIO_27/13-	15	G20	I/O	8	SLOW	
RIO_28/14+	15	H20	I/O	8	SLOW	
RIO_29/14-	15	H21	I/O	8	SLOW	
RIO_30/15+	15	L20	I/O	8	SLOW	
RIO_31/15-	15	L21	I/O	8	SLOW	
RIO_62/31+	15	G21	I/O	8	SLOW	
RIO_63/31-	15	G22	I/O	8	SLOW	
Reserved (Do Not Use!)						
Reserved	14	N17				Do Not Use!
Reserved	14	N20				Do Not Use!
Reserved	14	N21				Do Not Use!
Reserved	14	N22				Do Not Use!
Reserved	14	M19				Do Not Use!
Reserved	14	M21				Do Not Use!
Reserved	14	M22				Do Not Use!
Reserved	14	P20				Do Not Use!
Reserved	14	P21				Do Not Use!
Reserved	14	P22				Do Not Use!
Reserved	14	R16				Do Not Use!
Reserved	14	R17				Do Not Use!
Reserved	14	T17				Do Not Use!
Reserved	14	U17				Do Not Use!
Reserved	34	U8				Do Not Use!
Reserved	34	V8				Do Not Use!

Signal	Bank	Pin	Dir	Drive [mA]	Slew Rate	Comment
Reserved	34	W7				Do Not Use!
Reserved	34	W8				Do Not Use!
Reserved	34	Y4				Do Not Use!
Reserved	34	Y5				Do Not Use!
Reserved	34	Y6				Do Not Use!
Reserved	34	Y7				Do Not Use!
Reserved	34	Y8				Do Not Use!
Reserved	34	AA10				Do Not Use!
Reserved	35	H2				Do Not Use!
Reserved	35	H3				Do Not Use!
Reserved	35	H4				Do Not Use!
Reserved	35	J3				Do Not Use!
Reserved	35	M7				Do Not Use!

Table 7-3 : User FPGA I/O Signal to I/O Bank/Pin Mapping

7.5 Signal Description

In the table below, signal direction is from the User FPGA point of view

User FPGA I/O Signal	Direction	Description
Local Bus Signal Group		
LB_AD_0 ... 31	I/O	Local Bus Address/Data Bus See <i>Local Bus</i> chapter for a detailed description.
LB_CLK	I	Local Bus Clock See <i>Local Bus</i> chapter for a detailed description.
LB_RST#	I	Local Bus Reset See <i>Local Bus</i> chapter for a detailed description.
LB_CYC#	I	Local Bus Cycle Indication See <i>Local Bus</i> chapter for a detailed description.
LB_R/W#	I	Local Bus Read/Write Cycle See <i>Local Bus</i> chapter for a detailed description.
LB_D/A#	I	Local Bus Data/Address Phase Indication See <i>Local Bus</i> chapter for a detailed description.
LB_C/BE#_0 ... 3	I	Local Bus Control/ Byte Enable See <i>Local Bus</i> chapter for a detailed description.
LB_MRDY#	I	Local Bus Master Ready See <i>Local Bus</i> chapter for a detailed description.
LB_MLAST#	I	Local Bus Master Last Data Phase See <i>Local Bus</i> chapter for a detailed description.
LB_MABT#	I	Local Bus Master Abort See <i>Local Bus</i> chapter for a detailed description.
LB_TRDY#	O	Local Bus Target Ready See <i>Local Bus</i> chapter for a detailed description.
LB_TTERM#	O	Local Bus Target Termination See <i>Local Bus</i> chapter for a detailed description.
LB_TERR#	O	Local Bus Target Error See <i>Local Bus</i> chapter for a detailed description.

User FPGA I/O Signal	Direction	Description
LB_INT#	O, OD	Local Bus Interrupt See <i>Local Bus</i> Chapter for a detailed description.
Misc Signal Group		
BRG_CLK	I	Baud Rate Generator Clock (7.3728 MHz)
USR_CLK	I	General User Clock (40 MHz)
EMC_CLK	I	53.2 MHz \pm 5.5% (FPGA EMCCLK pin)
RIO_VCCO_SEL	O, OD	Rear I/O Bank Supply Select On-board pull-up resistor Not Used Weak Pull-Down High-Z Driven High: 3.3V Rear I/O (support for LVTTTL or LVCMOS33 I/O standards) Driven Low: 2.5V Rear I/O (required for LVCMOS25 and LVDS_25 I/O standards) CAUTION: See <i>Rear I/O Interface Signals</i> chapter! CAUTION: See <i>Handling and Operation Instructions</i> chapter!
ULED_1 ... 2	O	User LED Control 0: LED Off 1: LED On Note that the User LED control may be switched to the PCI Interface Device on certain conditions. See <i>LED Description</i> chapter.
BLD_OPT_0 ... 2	I	Build Option BLD_OPT[2:0]: 000: -10R Order/Build Option 001: -11R Order/Build Option 010: -12R Order/Build Option 011: -13R Order/Build Option 100: -14R Order/Build Option Others Reserved
I2C_SCL	OD	I2C EEPROM Serial Clock On-board pull-up resistor See <i>I2C EEPROM</i> chapter
I2C_SDA	I/OD	I2C EEPROM Serial Data On-board pull-up resistor See <i>I2C EEPROM</i> chapter
I/O Signal Group		
FIO_0 ... 63	I/O	Front I/O Data Signals for On-Board I/O Transceivers Input or Output direction depends on corresponding FIO_OE signal setting. Represents the I/O line level for Input Direction Sets the I/O line level for Output Direction The corresponding User FPGA FIO Output Buffer must be kept High-Z for Input Direction! On-board 100R series resistor for limited protection. Controlled/Distributed output switching circuitry and input filtering is recommended for single-ended I/O. Signal number matches the I/O channel number for single-ended I/O. Even signal numbers are used for the differential I/O channels (e.g. FIO[6] is used for differential I/O channel 3). See <i>Front I/O Interface Signals</i> chapter! See <i>Front I/O Interface</i> chapter!

User FPGA I/O Signal	Direction	Description
FIO_OE_0 ... 63	O	<p>Front I/O Line Direction Signals for On-Board I/O Transceivers</p> <p>On-Board pull-down resistor enforces Input Direction while the User FPGA is not configured</p> <p>0: Input Direction (Output Disabled)</p> <p>In Input Direction, the on-board I/O transceiver drives the I/O line level to the corresponding User FPGA FIO pin!</p> <p>The corresponding User FPGA FIO Output Buffer must be kept High-Z for Input Direction!</p> <p>1: Output Direction (Output Enabled)</p> <p>In Output Direction, the on-board I/O transceiver drives the corresponding User FPGA FIO signal to the I/O line.</p> <p>Signal number matches the I/O channel number for single-ended I/O. Even signal numbers are used for the differential I/O channels (e.g. FIO_OE[6] is used for differential I/O channel 3).</p> <p>See <i>Front I/O Interface Signals</i> chapter!</p> <p>See <i>Front I/O Interface</i> chapter!</p>
RIO_0/0+, RIO_1/0-, RIO_2/1+, RIO_3/1-, ... RIO_30/15+, RIO_31/15- RIO_62/31+, RIO_63/31-	I/O	<p>Rear I/O Lines</p> <p>Direct FPGA I/O pin connection to PMC P14</p> <p>No on-board termination nor line protection!</p> <p>[31:0] are available for all TPMC644 order options</p> <p>[63:62] are always available to support TCMC I2C management communication.</p> <p>Up to 32x Single-Ended (LVTTTL, LVCMOS33, LVCMOS25) or up to 16x Differential (LVDS_25) lines</p> <p>Differential requires a 2.5V Rear I/O bank supply (i.e. requires that RIO_VCCO_SEL is driven low) and LVDS_25 I/O standard for the differential lines and LVCMOS25 I/O standard for any single-ended I/O lines.</p> <p>See <i>Rear I/O Interface</i> chapter.</p> <p>CAUTION: See <i>Handling and Operation Instructions</i> chapter!</p>

Table 7-4 : User FPGA Signal Description

Note that the User FPGA I/O pins for front I/O are always single-ended signals in the User FPGA design and on-board I/O line transceivers are converting the User FPGA data signals to the differential front I/O line signals on the corresponding order options. The User FPGA I/O pins for rear I/O are directly connected to the P14 rear I/O connector and are either used as single-ended or as differential signals in the User FPGA design.

7.5.1 Clock Sources

Available clock signals (inputs to User FPGA) are: LB_CLK (Local Bus Clock), USR_CLK (User Clock), BRG_CLK (Baud Rate Generator Clock) and EMC_CLK (Clock that may be used for User FPGA Configuration from SPI Flash). Typically, LB_CLK is the only clock signal that is really required for a user design.

See User FPGA Signal Description table for more information.

7.5.2 Front I/O Interface Signals

7.5.2.1 Useable FIO & FIO_OE FPGA Pins

Note that the 64 FIO and 64 FIO_OE pins/signals on the User FPGA are always present and are always single-ended signals, even for order options providing differential front I/O signals. On-board transceivers are translating the single-ended FIO signals to differential front I/O signals on the corresponding order options.

Depending on the order option, not all of the 64 FIO and FIO_OE pins are useable (though they are always present on the User FPGA). E.g. the TPMC644-11R order option provides 32 differential I/O lines on the front I/O interface. Hence there are 32 on-board differential transceivers and only 32 of the 64 FIO and 32 of the 64 FIO_OE signals are useable on the TPMC644-11R.

The TPMC644-10R order option provides 64 single-ended front I/O signals and uses all 64 FIO and FIO_OE signals 0, 1, 2, 3, ..., 63 for the mapping.

The TPMC644-11R/-13R order options provide 32 differential I/O signals and uses 32 of the 64 FIO and FIO_OE signals. Only the even FIO and FIO_OE signals 0, 2, 4, ..., 62 are used.

The TPMC644-12R/-14R order options provide 16 differential plus 32 single-ended I/O signals. The even FIO and FIO_OE signals 0, 2, 4, ..., 30 are used for the differential I/O signals and the FIO and FIO_OE signals 32 ... 63 are used for the single-ended I/O signals.

The following tables are more or less showing the same information from different angles.

Front I/O Line Line (Channel)	User FPGA FIO & FIO_OE Signals for various Order Options		
	-10R	-11R, -13R	-12R, -14R
Single Ended I/O Lines 0 ... 15	0 ... 15	N/A	N/A
Single Ended I/O Lines 16 ... 31	16 ... 31	N/A	N/A
Single Ended I/O Lines 32 ... 47	32 ... 47	N/A	32 ... 47
Single Ended I/O Lines 48 ... 63	48 ... 63	N/A	48 ... 63
Differential I/O Lines 0+/- ... 15+/-	N/A	0 / 2 / 4 / ... / 30	0 / 2 / 4 / ... / 30
Differential I/O Lines 16+/- ... 31+/-	N/A	32 / 34 / 36 / ... / 62	N/A

Table 7-5 : Front I/O Line to User FPGA I/O Signal Mapping

User FPGA FIO & FIO_OE Signals	Front I/O Line (Channel) for various Order Options		
	-10R	-11R, -13R	-12R, -14R
0 ... 15	(Single-Ended) 0 ... 15	(Differential) 0 → 0+/- 2 → 1+/- 4 → 2+/- ... 14 → 7+/-	(Differential) 0 → 0+/- 2 → 1+/- 4 → 2+/- ... 14 → 7+/-
16 ... 31	(Single-Ended) 16 ... 31	(Differential) 16 → 8+/- 18 → 9+/- 20 → 10+/- ... 30 → 15+/-	(Differential) 16 → 8+/- 18 → 9+/- 20 → 10+/- ... 30 → 15+/-
32 ... 47	(Single-Ended) 32 ... 47	(Differential) 32 → 16+/- 34 → 17+/- 36 → 18+/- ... 46 → 23+/-	(Single-Ended) 32 ... 47
48 ... 63	(Single-Ended) 47 ... 63	(Differential) 48 → 24+/- 50 → 25+/- 52 → 26+/- ... 62 → 31+/-	(Single-Ended) 48 ... 63

Table 7-6 : User FPGA I/O Signal to Front I/O Line Mapping (I)

FIO, FIO_OE Signal	Front I/O Line (Channel) for various Order Options		
	-10R	-11R, -13R	-12R, -14R
FIO_0, FIO_OE_0	IO_0	IO_0+/-	IO_0+/-
FIO_1, FIO_OE_1	IO_1	N/A	N/A
FIO_2, FIO_OE_2	IO_2	IO_1+/-	IO_1+/-
FIO_3, FIO_OE_3	IO_3	N/A	N/A
FIO_4, FIO_OE_4	IO_4	IO_2+/-	IO_2+/-
FIO_5, FIO_OE_5	IO_5	N/A	N/A
FIO_6, FIO_OE_6	IO_6	IO_3+/-	IO_3+/-
FIO_7, FIO_OE_7	IO_7	N/A	N/A
FIO_8, FIO_OE_8	IO_8	IO_4+/-	IO_4+/-
FIO_9, FIO_OE_9	IO_9	N/A	N/A
FIO_10, FIO_OE_10	IO_10	IO_5+/-	IO_5+/-
FIO_11, FIO_OE_11	IO_11	N/A	N/A
FIO_12, FIO_OE_12	IO_12	IO_6+/-	IO_6+/-

FIO, FIO_OE Signal	Front I/O Line (Channel) for various Order Options		
	-10R	-11R, -13R	-12R, -14R
FIO_13, FIO_OE_13	IO_13	N/A	N/A
FIO_14, FIO_OE_14	IO_14	IO_7+/-	IO_7+/-
FIO_15, FIO_OE_15	IO_15	N/A	N/A
FIO_16, FIO_OE_16	IO_16	IO_8+/-	IO_8+/-
FIO_17, FIO_OE_17	IO_17	N/A	N/A
FIO_18, FIO_OE_18	IO_18	IO_9+/-	IO_9+/-
FIO_19, FIO_OE_19	IO_19	N/A	N/A
FIO_20, FIO_OE_20	IO_20	IO_10+/-	IO_10+/-
FIO_21, FIO_OE_21	IO_21	N/A	N/A
FIO_22, FIO_OE_22	IO_22	IO_11+/-	IO_11+/-
FIO_23, FIO_OE_23	IO_23	N/A	N/A
FIO_24, FIO_OE_24	IO_24	IO_12+/-	IO_12+/-
FIO_25, FIO_OE_25	IO_25	N/A	N/A
FIO_26, FIO_OE_26	IO_26	IO_13+/-	IO_13+/-
FIO_27, FIO_OE_27	IO_27	N/A	N/A
FIO_28, FIO_OE_28	IO_28	IO_14+/-	IO_14+/-
FIO_29, FIO_OE_29	IO_29	N/A	N/A
FIO_30, FIO_OE_30	IO_30	IO_15+/-	IO_15+/-
FIO_31, FIO_OE_31	IO_31	N/A	N/A
FIO_32, FIO_OE_0	IO_32	IO_16+/-	IO_32
FIO_33, FIO_OE_33	IO_33	N/A	IO_33
FIO_34, FIO_OE_34	IO_34	IO_17+/-	IO_34
FIO_35, FIO_OE_35	IO_35	N/A	IO_35
FIO_36, FIO_OE_36	IO_36	IO_18+/-	IO_36
FIO_37, FIO_OE_37	IO_37	N/A	IO_37
FIO_38, FIO_OE_38	IO_38	IO_19+/-	IO_38
FIO_39, FIO_OE_39	IO_39	N/A	IO_39
FIO_40, FIO_OE_40	IO_40	IO_20+/-	IO_40
FIO_41, FIO_OE_41	IO_41	N/A	IO_41
FIO_42, FIO_OE_42	IO_42	IO_21+/-	IO_42
FIO_43, FIO_OE_43	IO_43	N/A	IO_43
FIO_44, FIO_OE_44	IO_44	IO_22+/-	IO_44
FIO_45, FIO_OE_45	IO_45	N/A	IO_45
FIO_46, FIO_OE_46	IO_46	IO_23+/-	IO_46
FIO_47, FIO_OE_47	IO_47	N/A	IO_47
FIO_48, FIO_OE_48	IO_48	IO_24+/-	IO_48
FIO_49, FIO_OE_49	IO_49	N/A	IO_49
FIO_50, FIO_OE_50	IO_50	IO_25+/-	IO_50

FIO, FIO_OE Signal	Front I/O Line (Channel) for various Order Options		
	-10R	-11R, -13R	-12R, -14R
FIO_51, FIO_OE_51	IO_51	N/A	IO_51
FIO_52, FIO_OE_52	IO_52	IO_26+/-	IO_52
FIO_53, FIO_OE_53	IO_53	N/A	IO_53
FIO_54, FIO_OE_54	IO_54	IO_27+/-	IO_54
FIO_55, FIO_OE_55	IO_55	N/A	IO_55
FIO_56, FIO_OE_56	IO_56	IO_28+/-	IO_56
FIO_57, FIO_OE_57	IO_57	N/A	IO_57
FIO_58, FIO_OE_58	IO_58	IO_29+/-	IO_58
FIO_59, FIO_OE_59	IO_59	N/A	IO_59
FIO_60, FIO_OE_60	IO_60	IO_30+/-	IO_60
FIO_61, FIO_OE_61	IO_61	N/A	IO_61
FIO_62, FIO_OE_62	IO_62	IO_31+/-	IO_62
FIO_63, FIO_OE_63	IO_63	N/A	IO_63

Table 7-7 : User FPGA I/O Signal to Front I/O Line Mapping (II)

For TPMC644-11R, FIO_OE signals 1, 3, 5, ..., 63 should be left unused and FIO signals 1, 3, 5, ..., 63 should be assigned an FPGA internal weak pull-down.

For TPMC644-12R, FIO_OE signals 1, 3, 5, ..., 31 should be left unused and FIO signals 1, 3, 5, ..., 31 should be assigned an FPGA internal weak pull-down.

7.5.2.2 FIO & FIO_OE FPGA Pin Usage

There are two User FPGA I/O signals for each front I/O line: FIO_OE[i] (output enable aka direction) and FIO[i] (input or output data). The FIO_OE[i] signals feature an on-board pull-down resistor (default direction = input).

Typically, in the User FPGA design, all FIO_OE[i] pins/signals are always active outputs with default level 0 (input direction). A User FPGA FIO pin/signal output buffer must be disabled when the corresponding FIO_OE signal is set to 0 for input direction and must be enabled when the corresponding FIO_OE signal is set to 1 for output direction.

When changing the I/O direction, the User FPGA FIO[i] pin and the on-board I/O line receiver may be driving simultaneously against each other for a short time (driver contention!), which is a condition that should be avoided!

Recommended sequence for changing the I/O line direction from input to output:

- 1st Drive the User FPGA FIO_OE[i] pin high (the on-board line receiver stops driving towards the FPGA FIO[i] pin)
- 2nd Wait at least 30 ns
- 3rd Enable the User FPGA FIO[i] pin output buffer

Recommended sequence for changing the I/O line direction from output to input:

- 1st Disable the User FPGA FIO[i] pin output buffer
- 2nd Wait at least 30 ns
- 3rd Drive the User FPGA FIO_OE[i] pin low (the on-board line receiver starts driving towards the FPGA FIO[i] pin)

Front I/O direction switching guideline:

Typically the User FPGA FIO_OE pins always have their output buffer enabled (permanently setting the I/O direction), driving either a low (input) or high level (output) to the corresponding on-board transceiver direction pin(s). The User FPGA FIO pins must have their output buffer disabled for input direction or enabled for output direction.

Note that the on-board Front I/O transceivers are actively driving the I/O line signal level onto the corresponding User FPGA FIO pin when/while the corresponding User FPGA FIO_OE signal is setting the on-board transceivers to input direction. The User FPGA FIO pin output buffer must not be enabled while the corresponding User FPGA FIO_OE pin is set for input direction!

When switching a Front I/O line from Input direction to Output direction, first set the corresponding User FPGA FIO_OE signal for Output direction before enabling the User FPGA FIO pin output buffer!

When switching a Front I/O line from Output direction to Input direction, first disable the corresponding User FPGA FIO pin output buffer, then set the corresponding User FPGA FIO_OE signal for Input direction.

User FPGA Device			I/O Interface	Comment
FIO_OE[i]	FIO[i] Output Driver	FIO[i]	IO[i]	
0	Disabled	Same as IO[i]	0 (externally set)	I/O channel used in input direction. On-Board I/O line receiver drives I/O line level to the FIO[i] pin.
			1 (externally set)	
1	Enabled	0	FIO[i] Level is driven to I/O Line	I/O channel used in output direction. On-board I/O line transmitter drives the FIO[i] level to the I/O line.
		1		
1	Enabled	0	0 is driven to I/O Line	For Single-Ended Front I/O only! (Pull-Float Switching, e.g. for 5V CMOS levels) For Output Direction only!
0			On-board I/O line transmitter is High-Z. I/O line level is pulled to pull reference by on-board pull-resistor.	

Table 7-8 : User FPGA Front I/O Line Table

7.5.3 Rear I/O Interface Signals

The TPMC644 features 32 + 2 P14 Rear I/O signals, directly connected to dedicated User FPGA I/O pins.

For single-ended rear I/O, every User FPGA RIO signal represents a rear I/O signal.

For differential rear I/O, two corresponding User FPGA RIO signals are required per differential rear I/O pair.

The rear I/O signal names are called RIO_0/0+, RIO_1/0- and so on. As an example, the two User FPGA I/O pins RIO_0/0+ and RIO_1/0- could either be used as two single-ended rear I/O signals RIO_0 and RIO_1 or as a single differential pair rear I/O signal RIO_0+/-.

The FPGA bank supply for the rear I/O signals is user configurable to be either 3.3V (default) or 2.5V.

When configured to 3.3V, the following FPGA I/O standards are available for the rear I/O signals: LVTTTL, LVCMOS33.

When configured to 2.5V, the following FPGA I/O standards are available for the rear I/O signals: LVDS_25, LVCMOS25.

LVDS_25 is the only option for differential rear I/O. Differential rear I/O cannot be combined with 3.3V single-ended rear I/O (differential rear I/O can only be combined with 2.5V single-ended rear I/O).

The TPMC644 does not provide any on-board termination resistors for the rear I/O signals/lines. For LVDS_25, User FPGA internal termination must be used.

The User FPGA Rear I/O bank supply and signal level can be configured via the RIO_VCCO_SEL signal available on a User FPGA I/O pin. When the signal is not used, weakly pulled down, High-Z or driven high, the rear I/O signal supply/level is set to 3.3V. When the signal is driven low by the User FPGA design, the rear I/O signal supply/level is switched to 2.5V. The selection made by the User FPGA RIO_VCCO_SEL signal may also be overridden by software via the Board Control & Status Register in the PCI Interface Device Register Space.

Signal(s)	Dir	User FPGA RIO_VCCO_SEL Signal	User FPGA Bank Supply	Level, IO Standard	Description
RIO_i	I/O	Not Used, Weak Pull-Down, High-Z, Driven High	3.3V (Default)	LVCMOS33	Single-Ended Direct User FPGA I/O Signal NOT 5V TOLERANT!
	I/O	Driven Low	2.5V	LVCMOS25	Single-Ended Direct User FPGA I/O Signal NOT 3.3V TOLERANT!
RIO_i+, RIO_i-	I/O	Driven Low	2.5V	LVDS_25	Differential Direct User FPGA I/O Pin Pair NOT 3.3V TOLERANT!

Table 7-9 : Rear I/O Signal Description

Normally, there are only two scenarios for which the User FPGA design must drive the RIO_VCCO_SEL pin low (the pin should best be left totally unused in all other cases):

- If rear I/O signals are connected to external inputs that have a maximum high level of +2.5V (i.e. are not even 3.3V tolerant). In this case the User FPGA design must use the LVCMOS25 I/O standard for all single-ended rear I/O signals.
- If LVDS should be used on some or all of the rear I/O signals. In this case the User FPGA design must use the LVDS_25 I/O standard on the differential rear I/O signal pairs and LVCMOS25 on any remaining single-ended rear I/O signals.

See the CAUTION! notes on the following page!

CAUTION!

The Rear I/O signals are direct FPGA I/O pin signals without any additional on-board transceiver or protection!

When used with a 3.3V bank supply, applying external voltage levels > 3.6V to the rear I/O signals will damage the FPGA device!

When used with a 2.5V bank supply, applying external voltage levels > 2.625V to the rear I/O signals will damage the FPGA device!

7.6 User FPGA Design Hints

7.6.1 Single-Ended Front I/O

Note that available I/O cables (e.g. Tews Technologies TA-104) are typically optimized for differential signals and will have twisted pair cables inside, which is not really suitable for single-ended I/O signals.

To prevent or mitigate unwanted noise effects (e.g. crosstalk and ground bouncing), it is recommended to use distributed output line switching (not simultaneous output line switching) and to use input filtering for single-ended I/O signals in the User FPGA design!

The PMC shall always be mounted by screws to the carrier board for an optimized ground connection.

For TPMC644-11R, FIO_OE signals 1, 3, 5, ..., 63 should be left unused and FIO signals 1, 3, 5, ..., 63 should be assigned an FPGA internal weak pull-down.

For TPMC644-12R, FIO_OE signals 1, 3, 5, ..., 31 should be left unused and FIO signals 1, 3, 5, ..., 31 should be assigned an FPGA internal weak pull-down.

7.6.2 User EEPROM

The TPMC644 provides a 16 Kbyte (128 Kbit) M24128 I2C EEPROM whose SCL and SDA signals are available at User FPGA I/O pins. Both signals have an on-board pull-up resistor.

I2C bus access structure for the User EEPROM is:

Start Condition – Device Select Byte – Higher Address Byte (A15 ... A8) – Lower Address Byte (A7 ... A0) – Data Byte(s) – Stop Condition

The device select code is 0b1010001 for reads or 0b1010000 for writes.

16 Kbyte addressing requires address bits A13 ... A0. Address bits 15 and 14 are always 0.

Recommended maximum clock frequency is 400 kHz.

Please consult the M24128 data sheet for more information.

8 User FPGA Configuration

8.1 User FPGA Configuration Options

The User FPGA must be configured (programmed with a configuration data bitstream) after every power cycle (the User FPGA is SRAM based and loses its content when power is turned off).

The TPMC644 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 via the JTAG header and programmer cable.
- FPGA Configuration (volatile) 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 again.
- 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 of again.

After power-up, the Spartan-7 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 TPMC644 Spartan-7 Board Reference Design (BRD).

8.1.1 Configuration Data Files

Programming the SPI Flash (for non-volatile User FPGA configuration) or programming the User FPGA directly (volatile) in Slave SelectMAP configuration mode requires certain result files from the SW tool design flow. Typically the .bin or the .bit file could be used as a programming file. The .bit file is the standard generated programming file. This is a binary configuration data file which contains additional header information that does not need to be programmed to either SPI Flash or FPGA. The .bin file is also a binary configuration data file but without the additional header information. The .bin file generation must first be enabled (is not enabled per default). In general both files could be used for programming, however the .bin file is preferable.

The .bit and/or .bin file generation must correspond to the desired configuration method, i.e. the result file must be generated either for FPGA configuration from SPI flash or for direct FPGA configuration via Slave SelectMap.

A configuration data file generated for Master Serial SPI configuration cannot be used for Slave SelectMap configuration and vice versa.

The number of bytes that must be written corresponds to the size of the result files.

.bit File Example

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000010 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000020 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000030 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000040 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000050 xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx xx
00000070 xx xx xx xx xx xx xx xx xx xx xx xx xx xx FF FF
00000080 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
```

```

00000090  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000A0  00 bb 11 22 00 44 FF FF FF FF FF FF FF AA 99
000000B0  55 66 20 00 00 00 30 03 E0 01 00 00 0B 30 00
...
002172F0  00 00 20 00 00 00 20 00 00 00 20 00 00 20 00
00217300  00 00 20 00 00 00 20 00 00 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
00000010  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000020  00 00 00 bb 11 22 00 44 FF FF FF FF FF FF FF FF
00000030  AA 99 55 66 20 00 00 00 30 03 E0 01 00 00 0B
...
00217270  20 00 00 00 20 00 00 00 20 00 00 00 20 00 00 00
00217280  20 00 00 00 20 00 00 00 20 00 00 00

```

8.2 FPGA Configuration from SPI Flash

8.2.1 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. If the SPI Flash is not programmed with a valid bitstream, the User FPGA will remain un-configured.

Upon delivery, the SPI Flash contains a TPMC644 Spartan-7 Board Reference Design (BRD).

8.2.2 SW Controlled Re-Configuration

Besides a power-cycle, an FPGA re-configuration from the on-board SPI Flash can also be initiated via a PCI Interface Device register.

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-7 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-7 FPGA PROGRAM_B pin and will prepare the User FPGA for re-configuration in Master Serial / SPI configuration mode.
3. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until both bits 2 & 3 are clear

I.e. wait until the Spartan-7 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-7 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-7 FPGA asserts the DONE pin high, indicating successful FPGA re-configuration.

8.2.3 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.

Upon delivery, the SPI Flash contains a TPMC644 Spartan-7 Board Reference Design (BRD).

8.2.4 SPI Configuration Parameters

SPI Flash Bus Width

The TPMC644 supports x1 or x2 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 TPMC644 supports using the Spartan-7 internal CCLK oscillator ($\pm 50\%$, 60/40) as the configuration clock source.

For a more deterministic configuration, the TPMC644 also provides a 53.2 MHz ($\pm 5.5\%$) clock signal on the EMCCLK pin that may be used as the configuration clock source.

Setting	Allowed Values (Options)			
BITSTREAM.CONFIG.EXTMASTERCCLK_EN	Disable	Disable	DIV-2	DIV-4
BITSTREAM.CONFIG.CONFIGRATE	12	26	-	-
BITSTREAM.CONFIG.SPI_32BIT_ADDR	No			
BITSTREAM.CONFIG.SPI_BUSWIDTH	1, 2			
BITSTREAM.CONFIG.SPI_FALL_EDGE	No, Yes	Yes	Yes	No, Yes
CONFIG_VOLTAGE	3.3			
CFGBVS	VCCO			
CONFIG_MODE	SPIx1, SPIx2			
Approximate User FPGA Configuration Time (x1)	2.92 sec (max)	1.35 sec (max)	0.70 sec (max)	1.40 sec (max)

Table 8-1 : SPI Configuration Properties

8.3 FPGA Configuration via PCI/Software

The TPMC644 supports direct (volatile) FPGA In-System-Programming via the PCI bus in Slave SelectMAP FPGA configuration mode (x8). Note that direct (volatile) FPGA configuration gets lost when power is turned off again.

Registers used for direct (volatile) FPGA In-System-Programming are the User FPGA Configuration Control/Status Register and the User FPGA Configuration Data Register, both located in the PCI Interface Device Register Space in PCI BAR0.

Direct (volatile) User FPGA In-System-Programming via PCI/Software requires the ISP_EN bit in the ISP Control Register (0xE0) to be set.

8.3.1 Direct FPGA Programming

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

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

1. Write 0x00000001 to the ISP Control Register (0xE0)
This will establish the required configuration path signal connections.
2. Write 0x00000003 to the User FPGA Configuration Control/Status Register (0xD0)
This will assert the Spartan-7 FPGA PROGRAM_B pin and will prepare a FPGA re-configuration in Slave SelectMAP configuration mode.
The Spartan-7 User FPGA will respond by driving both the DONE and INIT_B pins low.
3. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until bits 2 & 3 are clear
I.e. wait until the Spartan-7 FPGA asserts the INIT_B and DONE pins low.
4. Write 0x00000001 to the User FPGA Configuration Control/Status Register (0xD0)
This will release the Spartan-7 FPGA PROGRAM_B pin and will start a FPGA re-configuration in Slave SelectMAP configuration mode.
The Spartan-7 FPGA will respond by releasing the INIT_B pin.
5. Keep reading the User FPGA Configuration Control/Status Register (0xD0) until bit 3 is set
I.e. wait until the Spartan-7 FPGA releases the INIT_B pin.
6. 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-7 User FPGA.
7. Write eight more times (0xFFFFFFFF) to the User FPGA Configuration Data Register (0xD4)
This will provide additional configuration clock cycles for the Spartan-7 FPGA configuration port which may be required (depending on the project's start-up settings).
8. Wait for approx. 200ms
9. Read the User FPGA Configuration Control/Status Register (0xD0) and check if bit 2 is set
I.e. check successful Spartan-7 FPGA configuration (DONE pin assertion).
10. Write eight more times (0xFFFFFFFF) to the User FPGA Configuration Data Register (0xD4)
This will provide additional configuration clock cycles for the Spartan-7 FPGA configuration port which may be required (depending on the SW tool & design 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-7 User FPGA Slave SelectMAP configuration port (operating in x8 port configuration) one after the other. The byte order sent to the Spartan-7 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.

The ISP Control Register (0xE0) must be cleared again, if the User FPGA should re-configure from the on-board SPI Flash.

Since the Spartan-7 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-7 User FPGA configuration port. Therefore, the source configuration data set must be non-bit-swapped by the user. It is recommended to use the AMD .BIN file format as the configuration data source file. BIN file generation must be enabled in the AMD SW design flow.

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

8.4 FPGA Configuration via JTAG Connector

The Spartan-7 FPGA always supports (direct, volatile) configuration via its JTAG port. The Spartan-7 JTAG port is accessible via the TPMC644 JTAG connector. The Spartan-7 User FPGA is the only device in the JTAG chain provided via the TPMC644 JTAG connector.

The TPMC644 JTAG connector directly supports the 14 pos. flat ribbon cable of the AMD JTAG Programmer.

9 SPI Flash Programming

9.1 SPI Flash Notes

The TPMC644 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-7 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 TPMC644 Spartan-7 User FPGA Board Reference Design.

9.1.1 SPI Flash Device Type

The TPMC644 SPI Flash Key Parameters are:

- Device IS25LP064
- 64Mbit (8Mbyte) Memory
- 32768 pages a 256 byte
Page base addresses are 0x000000, 0x000100, 0x000200, ..., 0x7FFF00
- 2048 sectors a 4096 byte (16 pages)
Sector base addresses are 0x000000, 0x001000, 0x002000, ..., 0x7FF000
- Address range 0h ... 7FFFFFFh

9.1.2 SPI Flash Programming Options

The TPMC644 provides the following options for programming the User FPGA SPI Flash:

- SPI Flash Programming via PCI/Software
- SPI Flash Programming via JTAG Header and AMD Software & Cable

9.2 SPI Flash Programming via PCI/Software

The TPMC644 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 x2 mode.

PCI BAR 0 (PCI Interface Device 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 again when the SPI Flash programming is done. The ISP_EN bit must be clear for User FPGA (re-) configuration from the SPI Flash.

9.2.1 SPI Flash Program Data

See sub-chapter *Configuration Data Files*.

9.2.2 SPI Flash Instructions

The following SPI Flash instructions are supported:

- Chip Erase (0x60)
- Sector Erase (0x20)
- Page Program (0x02)
- Page Read (x03)

Program and Read instructions are processed on a per-page base (page-by-page). The SPI Flash page size is 256 Byte.

9.2.3 Steps for SPI Flash Chip Erase Operation

1. Write 0x00000001 to the ISP Control Register (0xE0)
This will establish the required configuration path signal connections.
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 (re-) set the on-board configuration signal path for Spartan-7 User FPGA configuration from the on-board SPI Flash.

9.2.4 Steps for SPI Flash Sector Erase Operation

1. Write 0x00000001 to the ISP Control Register (0xE0)
This will establish the required configuration path signal connections.
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 0x7FF000) write 0x00F07F20 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 (re-) set the on-board configuration signal path for Spartan-7 User FPGA configuration from the on-board SPI Flash.

9.2.5 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 establish the required configuration path signal connections.
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 (re-) set the on-board configuration signal path for Spartan-7 User FPGA configuration from the on-board SPI Flash.

9.2.6 Steps for SPI Flash Read Operation

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

This will establish the required configuration path signal connections.
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 0x7FFF00) write 0x00FF7F03 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 (re-) set the on-board configuration signal path for Spartan-7 User FPGA configuration from the on-board SPI Flash.

9.3 SPI Flash Programming via JTAG Header

Programming the SPI Flash via the TPMC644 JTAG header requires the AMD software and JTAG Programmer with the 2mm pitch 14 pos. flat ribbon cable.

The AMD software supports indirect SPI Flash programming by loading (and running) a special SPI core/master logic into the Spartan-7 FPGA that stimulates the FPGA I/O pins connected to the SPI Flash as required by the SPI Flash programming protocol. Any User FPGA configuration formerly stored in the SPI Flash would be overwritten by the SPI core/master logic.

Programming the SPI Flash via the TPMC644 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.

10 Local Bus

10.1 Local Bus Interface Notes

The TPMC644 provides an on-board off-chip local bus for accessing the User FPGA as a target from the PCI bus. The TPMC644 local bus is clock 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 provides a 32-bit multiplexed address/data bus. Each local bus cycle consists of an address phase and a data phase (note: future implementations may support multiple data phases). For each optional user space (PCI BARs 2 to 5) there is a dedicated user space select encoding in the upper address lines during the address phase. (Lower) Address lines used for decoding within the space range are passed from the PCI bus to the local bus. (Upper) Address lines used for decoding/selecting the PCI BAR are passed as 0 to the local bus. Address lines LB_AD[1:0] are always "00" during the address phase.

PCI data byte lanes are mapped 1:1 to the local bus data byte lanes, e.g. PCI_AD[7:0] are 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 lock state.

10.2 Local Bus Cycle Description

For a local bus cycle the PCI Interface Device is acting as the local bus master while the User FPGA is acting as a local bus target.

During/throughout a local bus cycle the master asserts a cycle signal and a direction signal. When the cycle signal is not active, the local bus is considered to be in an idle phase (and aside from the local clock and reset signals, the other local bus signals should be considered invalid).

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 (note: future implementations may support multiple data phases). 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 a User Space Select encoding in the upper address lines during/throughout the address phase and 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 (i.e. the address phase or the data phase) are done when both the master and the target are indicating phase ready status. Not indicating phase ready status would insert wait states to the current local bus cycle phase. Once asserted for a local bus cycle phase, a ready indication must be kept until the phase is done.

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 (and the local bus cycle) is done when both the master and the target are indicating ready status.

There always is at least one Bus Idle Cycle between Local Bus Cycles.

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

The TPMC644 provides a local bus time-out monitor. The local bus time-out function is enabled by default and can 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 (approx. after 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 (LB_MABT# is asserted 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 single 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.

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

10.2.2 Local Bus Target Error

The TPMC644 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
- Non-existing local 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 on the PCI bus in any other way.

10.3 Local Bus Interface Signal Description

The PCI Interface Device is the Local Bus Master (M). The User FPGA is the Local Bus Target (T).

User FPGA I/O Signal	Signal Direction		Description
	Signal Flow	User FPGA View	
<p>General Notes:</p> <p>The following signals are only valid when LB_CYC# = 0: LB_R/W#, LB_D/A#, LB_MRDY#, LB_MLAST#, LB_MABT#, LB_TRDY#, LB_TTERM#, LB_TERR#</p> <p>Valid Address condition: LB_CYC# = 0 AND LB_D/A# = 0 AND LB_MRDY# = 0</p> <p>Vaid Write Data condition: LB_CYC# = 0 AND LB_R/W# = 0 AND LB_D/A# = 1 AND LB_MRDY# = 0</p> <p>Address or Data Phase is done when both LB_MRDY# = 0 AND LB_TRDY# = 0</p>			
LB_AD_0 ... 31	M <> T	I/O	<p>Local Bus Address/Data Bus (Multiplexed)</p> <p>Address Phase (LB_D/A# = 0): The master drives the address and local space select encoding (valid when LB_MRDY# = 0). LB_AD[29:28]: Local Space Select 00: Local Space 0 (PCI BAR2) 01: Local Space 1 (PCI BAR3) 10: Local Space 2 (PCI BAR4) 11: Local Space 3 (PCI BAR5) LB_AD[27:2]: Local Address (max 256 Mbyte)</p> <p>Data Phase (LB_D/A# = 1): Writes: Master drives write data (valid when LB_MRDY# = 0) Reads: Target drives read data (valid when LB_TRDY# = 0) Note that for reads, the target shall insert a turn-around clock cycle for the multiplexed address/data bus. I.e. the target should not drive the data bus in the clock cycle immediately following the address phase.</p>
LB_CLK	M → T	I	<p>Local Bus Clock</p> <p>A buffered version of the PCI clock signal (max. 33 MHz)</p>
LB_RST#	M → T	I	<p>Local Bus Reset</p> <p>0: Local bus reset active 1: Local bus operating</p> <p>Active when/while:</p> <ul style="list-style-type: none"> • PCI Reset • User FPGA unconfigured (DONE = 0) • Local Bus SW Reset • Local Bus Clock PLL not locked
LB_CYC#	M → T	I	<p>Local Bus Cycle Indication</p> <p>0: Local bus cycle active (Address or Data Phase) 1: Local bus idle</p> <p>The following signals are only valid when LB_CYC# = 0: LB_R/W#, LB_D/A#, LB_MRDY#, LB_MLAST#, LB_MABT#, LB_TRDY#, LB_TTERM#, LB_TERR#</p>
LB_R/W#	M → T	I	<p>Local Bus Read/Write Cycle</p> <p>0: Write cycle 1: Read cycle</p>
LB_D/A#	M → T	I	<p>Local Bus Data/Address Phase Indication</p> <p>0: Address Phase 1: Data Phase</p>

User FPGA I/O Signal	Signal Direction		Description
	Signal Flow	User FPGA View	
LB_C/BE#_0 ... 3	M → T	I	<p>Local Bus Control/ Byte Enable</p> <p>Address Phase: Control information (reserved)</p> <p>Data Phase: 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</p> <p>0: Byte lane active 1: Byte lane not active</p> <p>LB_BE#_0 → AD[7:0] LB_BE#_1 → AD[15:8] LB_BE#_2 → AD[23:16] LB_BE#_3 → AD[31:24]</p>
LB_MRDY#	M → T	I	<p>Local Bus Master Ready</p> <p>Address Phase: 0: Address on local address/data bus is valid 1: Address on local address/data bus is not valid</p> <p>Data Phase: 0 (Write): Write data is valid on local address/data bus 0 (Read): Master is ready to sample read data from local address/data bus 1: Master is not ready to present write data on or sample read data from local address/data bus</p> <p>From the master point of view, LB_TRDY# and/or LB_TTERM# are only evaluated when LB_MRDY# is active.</p>
LB_MLAST#	M → T	I	<p>Local Bus Master Last Data Phase</p> <p>Only relevant for data phase(s) (high during address phase). By assertion the master indicates that the current data phase is the last data phase of the local bus cycle. The current implementation only supports a single data phase and this signal is always active during the (one and only) data phase. Future implementations may support multiple data phases per local bus cycle (bursts) and LB_MLAST# held high would indicate that the master attempts to perform another (next) data phase immediately after the current one.</p>
LB_MABT#	M → T	I	<p>Local Bus Master Abort</p> <p>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 a single clock cycle and will terminate and discard the current local bus cycle.</p>

User FPGA I/O Signal	Signal Direction		Description
	Signal Flow	User FPGA View	
LB_TRDY#	T → M	O	<p>Local Bus Target Ready</p> <p>Once asserted, the signal must stay asserted until the master indicates master ready state.</p> <p>Address Phase:</p> <p>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</p> <p>The Target may use 0 as the signal default level when always ready to sample the address coming from idle. This would allow a single clock cycle address phase.</p> <p>Data Phase:</p> <p>0 (Read): Target read data is valid on local address/data bus Note that the target shall insert a turn-around state right after the address phase, before driving the read data onto the AD bus. 0 (Write): Target is ready to sample write data from local address/data bus 1: Target is not ready to present read data or accept write data</p>
LB_TTERM#	T → M	O	<p>Local Bus Target Termination</p> <p>Only relevant in local bus cycle data phase(s).</p> <p>Intended for target termination of local bus burst cycles with multiple data phases in future implementations. By asserting this signal, the target forces that the current data phase is the last data phase of the local bus cycle (even if the master intends to have more data phases after the current one).</p> <p>The current implementation only uses single data phase local bus cycles and the target should assert LB_TTERM# in the same way as (equal to) LB_TRDY#.</p>
LB_TERR#	T → M	O	<p>Local Bus Target Error</p> <p>The target may report an error condition during a local bus cycle by driving this signal low for a single local clock cycle.</p> <p>In case of a target error indication during a local bus cycle, the master will terminate and discard the local bus cycle.</p>
LB_INT#	T → M	O, OD	<p>Local Bus Interrupt</p> <p>The target may assert a level sensitive interrupt by driving this signal low</p> <p>This signal has an on-board pull-up resistor</p> <p>May be used as open-drain or push-pull output</p> <p>0: Interrupt active 1, Z: Interrupt not active</p> <p>The Local Bus Interrupt may be masked by SW via PCI Interface Device registers (PCI BAR0).</p>

Table 10-1 : Local Bus Interface Signal Description

10.3.1 LB_TTERM# Notes (for Future Implementations)

See the Local Bus Interface Signal Description table for the current firmware implementation.

Future firmware implementations may support local bus burst cycles with multiple data phases following the address phase.

10.3.2 Local Bus Signal States

Shaded items are not in the current implementation.

Local Bus Master Signal States (Data Phase)		
LB_MRDY#	LB_MLAST#	Master Signal State
H	X	Master Wait State
L	H	Not present in the current implementation. Master Ready for current Data Phase, more Data Phases intended
L	L	Master Ready for current and last Data Phase (Master Completion)

Table 10-2 : Local Bus Master Signal States (Data Phase)

Local Bus Target Signal States (Data Phase)		
LB_TRDY#	LB_TTERM#	Target Signal State
H	H	Target Wait State
H	L	Current implementation: Same as Target Wait State
L	H	Current implementation: Same as Target Termination with Data
L	L	Target Termination with Data

General Note: These signals are evaluated by the master when LB_MRDY# is active.
Recommendation for current implementation: The target should assert LB_TTERM# in the same way as (equal to) LB_TRDY#.

Table 10-3 : Local Bus Target Signal States (Data Phase)

Local Bus Signal States (Data Phase)		
Master Signal State	Target Signal State	Local Bus Signal State
Master Wait State	Target Wait State	Wait State (Both)
	Target Termination without Data	Wait State (Master)
	Target Ready	Wait State (Master)
	Target Termination with Data	Wait State (Master)
Master Ready and Last Data Phase	Target Wait State	Wait State (Target)
	Target Termination without Data	Cycle Termination without Data Transfer (Target Termination)
	Target Ready	Cycle Termination with Data Transfer (Master Completion)
	Target Termination with Data	Cycle Termination with Data Transfer (Master & Target Completion)

Table 10-4 : Local Bus Signal States

10.4 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#, LB_TTERM#	10 ns
Local Bus Master Output Delay Timing	
LB_AD[31:0]	11 ns
LB_C/BE#[3:0], LB_CYC#, LB_R/W#, LB_D/A#, LB_MRDY#, LB_MABT#, LB_MLAST#	11 ns
Recommended Local Bus Target Input Setup Timing	
LB_AD[31:0]	10 ns
LB_C/BE#[3:0], LB_CYC#, LB_R/W#, LB_D/A#, LB_MRDY#, LB_MABT#, LB_MLAST#	10 ns
Recommended Local Bus Target Output Delay Timing	
LB_AD[31:0]	15 ns
LB_TRDY#, LB_TERR#, LB_TTERM#	12 ns

Table 10-5 : Local Bus Interface Timing

10.5 Local Bus Signal Protocol Example Diagrams

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

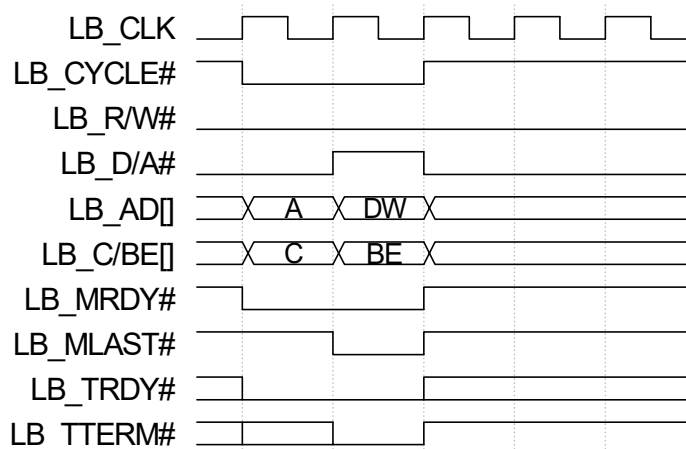


Figure 10-1 : Single Write Access

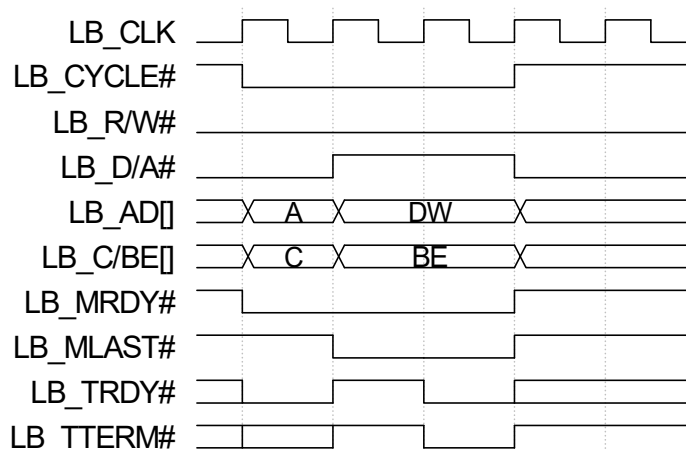


Figure 10-2 : Single Write Access (w/ Wait State)

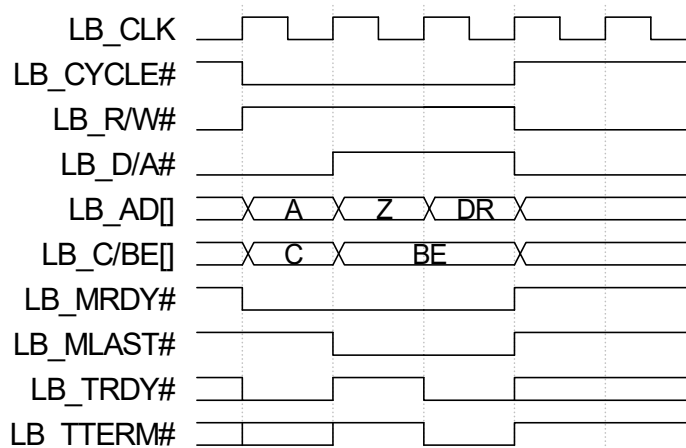


Figure 10-3 : Single Read Access

Note that for a read cycle, the target shall insert a turn-around cycle right after the address phase before starting to drive the read data! This avoids driver contention on the multiplexed address/data bus.

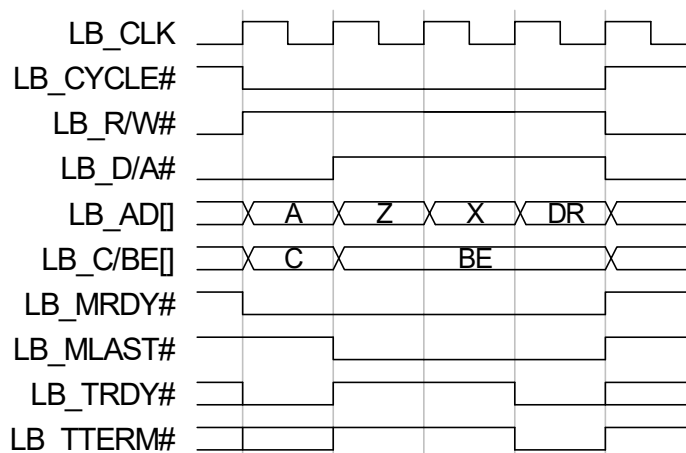


Figure 10-4 : Single Read Access (w/ Wait State)

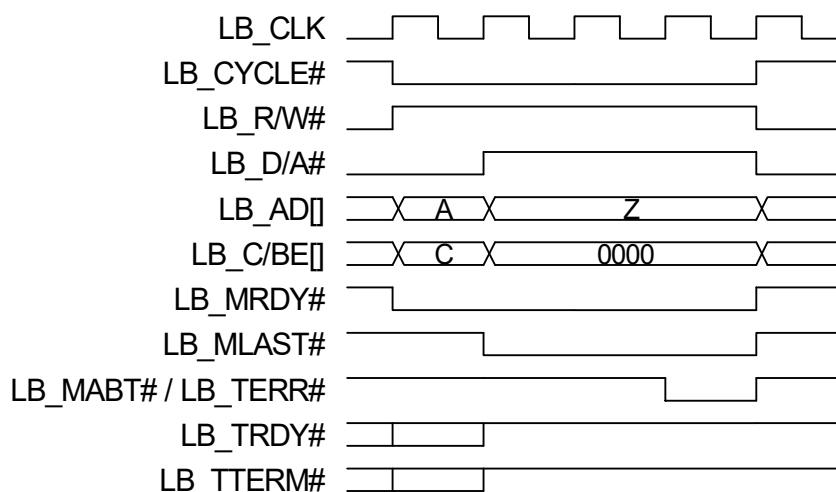


Figure 10-5 : Read with Master Abort (or Target Error)

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

LB_R/W#, LB_D/A#, LB_MRDY#, LB_MLAST#, LB_MABT#, LB_TRDY#, LB_TTERM#, 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 next local bus cycle may show a different master or target behavior.

11 Configuration Hints

11.1 Jumper Configuration

TPMC644 order options -10R, -12R and -14R are featuring single-ended front I/O lines.

The single-ended front I/O lines have individual pull resistors to a common on-board reference. The common on-board pull resistor reference is configurable by jumper J1 as described below.

Note that jumper J1 is not present on the TPMC644 differential front I/O only order options -11R and -13R.

On the mixed order options (-12R, -14R) the jumper is only relevant for the single-ended front I/O lines.

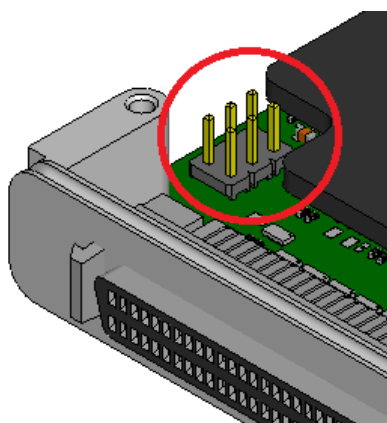


Figure 11-1 : Jumper Location

Jumper Setting	I/O Line Pull-Resistor Reference	Description
<p>3V3 REF 5V</p>	3.3V	10K Pull-Up to +3.3V for each Single-Ended I/O Line
<p>3V3 REF 5V</p>	5V	10K Pull-up to +5V for each Single-Ended I/O Line
<p>3V3 REF 5V</p>	GND	10K Pull-Down to 0V for each Single-Ended I/O Line

Table 11-1 : Jumper Configuration

Since the jumper configuration defines the same common pull-reference for all single-ended front I/O lines, in case of an on-board pull-up configuration, the user may provide external 1K resistors for individual single-ended front I/O lines that should differ in their default state.

12 Thermal Considerations



Forced air cooling is required for ambient temperatures exceeding +25°C! Without forced air cooling, damage to the device can occur. To avoid permanent damage of the User FPGA, the temperature of the Spartan-7 should be controlled/verified for a given User FPGA design and application environment (ambient temperature, forced air cooling, etc.)

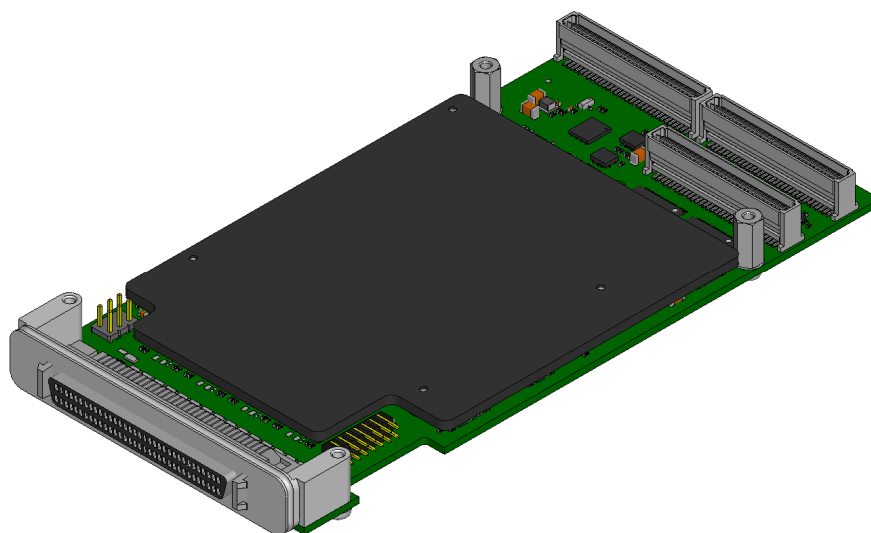


Figure 12-1 : TPMC644-10R with Heatsink

All components on the TPMC644 are rated for the industrial temperature range of -40°C to +85°C.

However, the maximum allowed ambient operating temperature highly depends on the User FPGA design, the performance and load of the I/O circuitry and the applied cooling method.

The absolute maximum junction temperature of the Spartan-7 User FPGA is +125°C and must never be exceeded!

The recommended maximum junction temperature of the Spartan-7 User FPGA is +100°C and should not be exceeded.

A simple system air cooling is not sufficient for heavy utilization of the User FPGA!

The user must use appropriate measures to ensure that the Spartan-7 User FPGA junction temperature stays within the given limits for the actual User FPGA logic design and environment/system conditions (ambient temperature, forced air cooling etc.).

13 I/O Interface Description

13.1 Overview

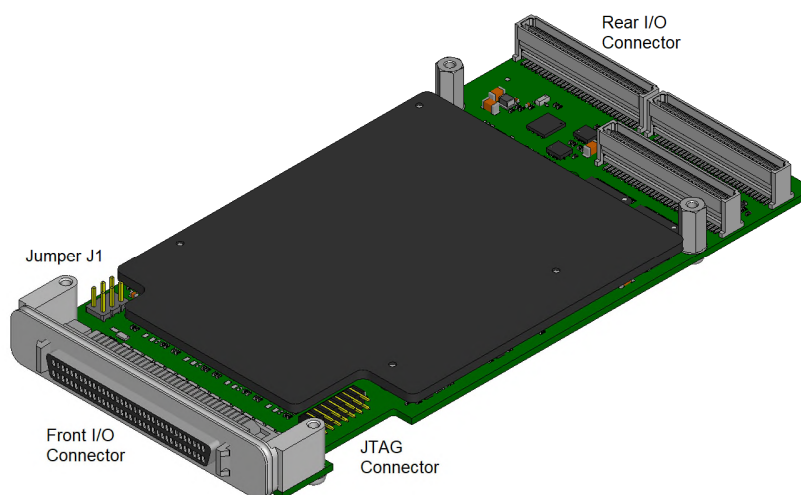


Figure 13-1 : TPMC644-10R Board Overview

13.2 LED Description

The TPMC644 features four LEDs, available on the bottom side of the card at a card edge.

The PG LED is controlled by the board hardware, the MXO LED is controlled by the PCI Interface Device logic.

The user LEDs USR1 and USR2 are typically controlled by the User FPGA design/logic. However, when the ISP mode is set active (via the PCI Interface Device registers), both user LEDs are forced On. The PCI Interface Device could also take over the user LED control via register programming.

LED	Color	State	Description
PG	Green	On	Power-Good indication
MXO	Green	On	Ready
		Flashing	Initializing or ISP mode active
USR1	Green	On	<ul style="list-style-type: none"> ISP mode is active Set under user control (either via the User FPGA design or via PCI Interface Device register). Factory default function see BRD documentation
USR2	Green	On	<ul style="list-style-type: none"> ISP mode is active Set under user control (either via the User FPGA design or via PCI/Local Bridge register). Factory default function see BRD documentation

ISP: In-System-Programming, BRD: Board Reference (User FPGA) Design

Table 13-1 : LED Description

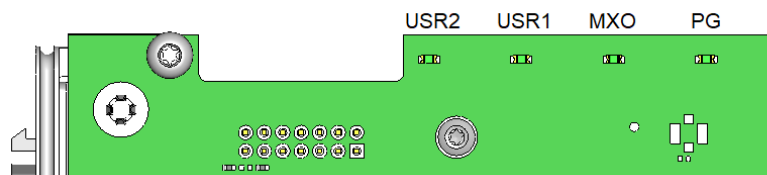


Figure 13-2 : LED Location

13.3 JTAG Connector

The JTAG connector provides exclusive access to the Spartan-7 User FPGA JTAG port.

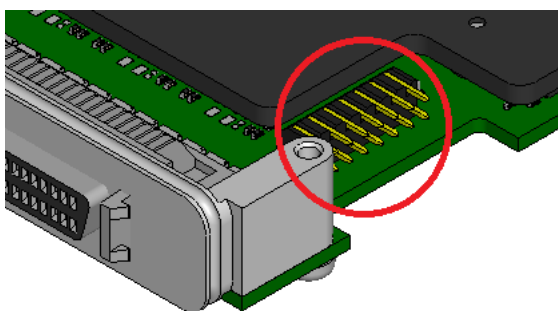


Figure 13-3 : JTAG Connector Location

The TPMC644 JTAG header directly supports plugging the AMD 14-pin 2 mm Pitch JTAG Programmer Cable.

13.3.1 Connector Type

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

Table 13-2 : JTAG Header

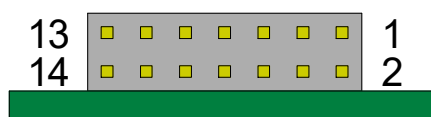


Figure 13-4 : JTAG Connector Pin-Out

13.3.2 Pin Assignment

Pin	Signal	Dir	Level	Description
1	NC			Not connected
2	+3.3V	From Card	+3.3V	+3.3V JTAG Signal Reference Supply
3	GND			Ground
4	JTAG_TMS	To Card	+3.3V	Test Mode Select
5	GND			Ground
6	JTAG_TCK	To Card	+3.3V	Test Clock
7	GND			Ground
8	JTAG_TDO	From Card	+3.3V	Test Data Output
9	GND			Ground
10	JTAG_TDI	To Card	+3.3V	Test Data Input
11	JTAG_PRSENT#	To Card	+3.3V	Indicates Programmer Present when externally tied low
12	NC			Not connected
13	RSV			Reserved
14	NC			Not connected

Table 13-3 : JTAG Connector Pin Assignment

13.4 Front I/O Interface

The number of available LVTTTL/TTL, RS422/584 or M-LVDS front I/O lines depends on the TPMC644 order option.

Order Option	Single-Ended I/O	Differential I/O	
	LVTTTL/TTL I/O Channels	RS422/485 I/O Channels	M-LVDS I/O Channels
-10R	64	-	-
-11R	-	32	-
-12R	32	16	-
-13R	-	-	32
-14R	32	-	16

Table 13-4 : Order Option Front I/O Overview

It is important to understand, that the front I/O lines are not directly connected to User FPGA I/O pins. On-board I/O transceiver devices are used for the front I/O interface and the corresponding User FPGA I/O pins are connected to the system side of the I/O transceivers while the line side of the I/O transceivers is connected to the front I/O connector.

Also see the *Front I/O Interface Signals* section in the *User FPGA* chapter for more information.

13.4.1 Single-Ended I/O (LVTTTL)

TPMC644 order options -10R, -12R and -14R are featuring LVTTTL single-ended I/O lines.

Each single-ended I/O line can be either configured/used as input or output. Default configuration is input.

Each single-ended I/O line features a small series damping resistor and an on-board pull-resistor (always present and connected to the on-board pull reference). There is a common on-board pull reference for all single-ended I/O lines which is configurable by jumper. Configurable pull reference options are: 3.3V, 5V or Ground (or floating). See *Jumper Configuration* chapter for the jumper settings.

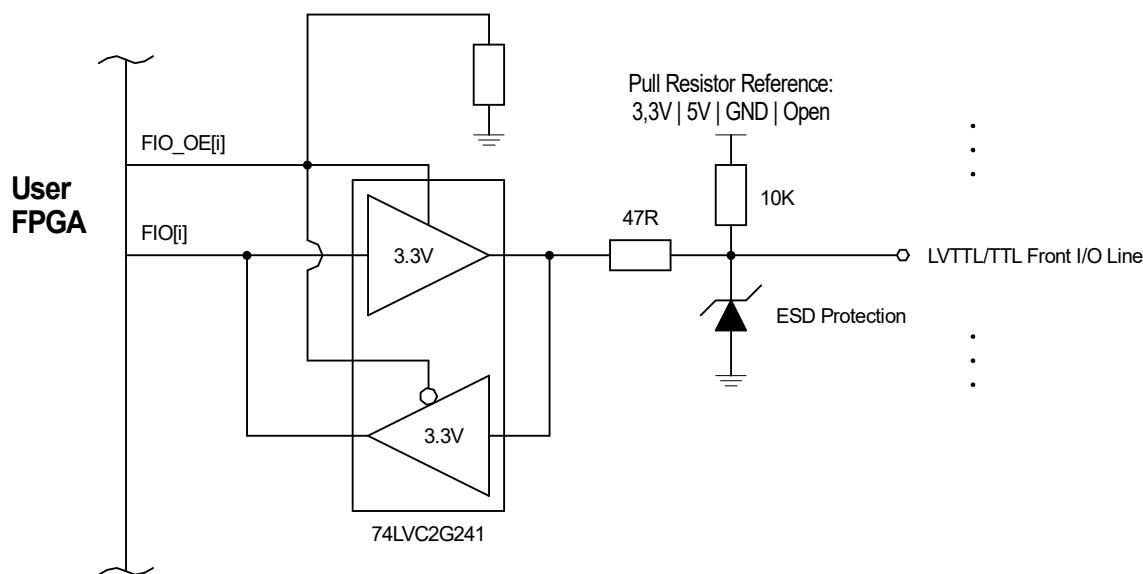


Figure 13-5 : Single-Ended Front I/O Line Circuit

The single-ended I/O transceiver operates from a 3.3V supply, however the single-ended inputs are 5V tolerant.

As an output, the single-ended transmitter drives either low level or 3.3V high level.

5V (CMOS) output level can be reached (more in a static way) by using the 5V pull reference. For this, the FPGA FIO (data) signal is set to output direction with a constant low level and the front I/O line state/level is set via the FPGA FIO_OE (output enable) signal (inverted). However this is only supported for slow signals and external high-impedance loads ($> 25K$).

13.4.2 Differential I/O (RS422/485, M-LVDS)

TPMC644 order options -11R, -12R, -13R and -14R are featuring differential I/O lines.

Each differential I/O line can be either configured/used as an input or as an output. Default configuration is input.

Each differential I/O line features an on-board differential termination resistor. The termination resistor value is 120Ω for RS422/485 and 100Ω for M-LVDS order options.

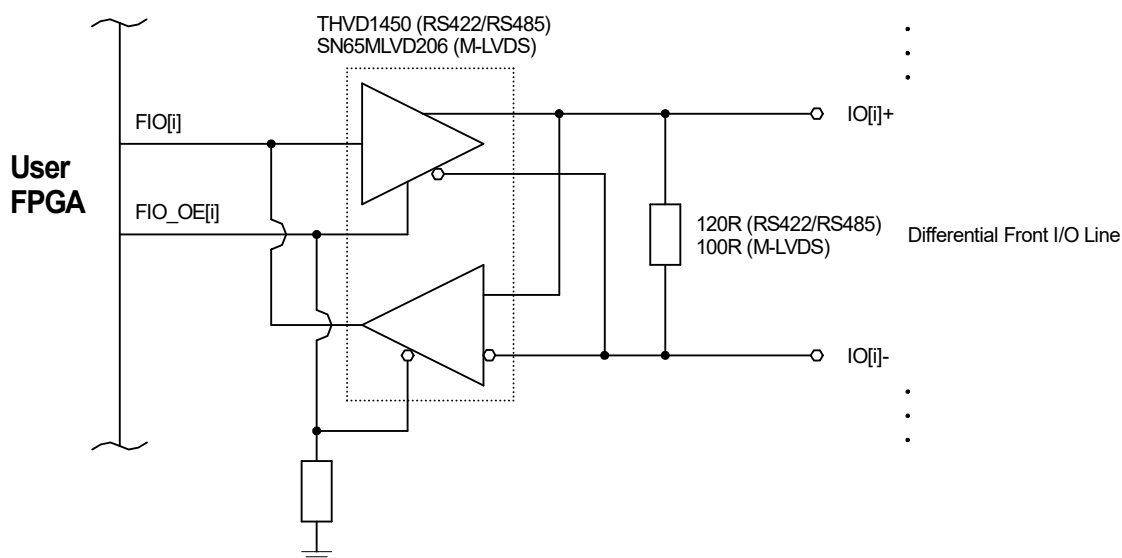


Figure 13-6 : Differential Front I/O Line Circuit

13.5 Front I/O Connector

13.5.1 Connector Type

Pin-Count	68
Connector Type	HD68 / SCSI-3 Connector
Source & Order Info	AMP 5-787082-7 or compatible

Table 13-5 : Front I/O Connector Type

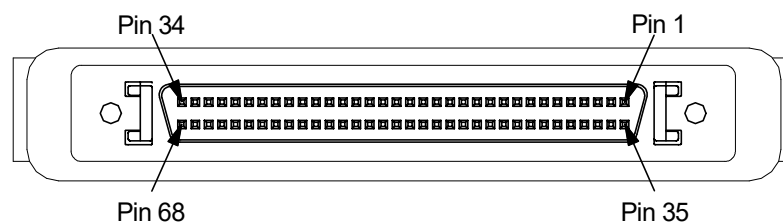


Figure 13-7 : Front I/O Connector Pin-Out

13.5.2 Front I/O Connector Pin Assignment TPMC644-10R

Pin	Signal	Dir	Level	Pin	Signal	Dir	Level
1	IO_0	I/O	LVTTL/TTL	35	IO_1	I/O	LVTTL/TTL
2	IO_2	I/O	LVTTL/TTL	36	IO_3	I/O	LVTTL/TTL
3	IO_4	I/O	LVTTL/TTL	37	IO_5	I/O	LVTTL/TTL
4	IO_6	I/O	LVTTL/TTL	38	IO_7	I/O	LVTTL/TTL
5	IO_8	I/O	LVTTL/TTL	39	IO_9	I/O	LVTTL/TTL
6	IO_10	I/O	LVTTL/TTL	40	IO_11	I/O	LVTTL/TTL
7	IO_12	I/O	LVTTL/TTL	41	IO_13	I/O	LVTTL/TTL
8	IO_14	I/O	LVTTL/TTL	42	IO_15	I/O	LVTTL/TTL
9	GND			43	GND		
10	IO_16	I/O	LVTTL/TTL	44	IO_17	I/O	LVTTL/TTL
11	IO_18	I/O	LVTTL/TTL	45	IO_19	I/O	LVTTL/TTL
12	IO_20	I/O	LVTTL/TTL	46	IO_21	I/O	LVTTL/TTL
13	IO_22	I/O	LVTTL/TTL	47	IO_23	I/O	LVTTL/TTL
14	IO_24	I/O	LVTTL/TTL	48	IO_25	I/O	LVTTL/TTL
15	IO_26	I/O	LVTTL/TTL	49	IO_27	I/O	LVTTL/TTL
16	IO_28	I/O	LVTTL/TTL	50	IO_29	I/O	LVTTL/TTL
17	IO_30	I/O	LVTTL/TTL	51	IO_31	I/O	LVTTL/TTL
18	IO_32	I/O	LVTTL/TTL	52	IO_33	I/O	LVTTL/TTL
19	IO_34	I/O	LVTTL/TTL	53	IO_35	I/O	LVTTL/TTL
20	IO_36	I/O	LVTTL/TTL	54	IO_37	I/O	LVTTL/TTL
21	IO_38	I/O	LVTTL/TTL	55	IO_39	I/O	LVTTL/TTL
22	IO_40	I/O	LVTTL/TTL	56	IO_41	I/O	LVTTL/TTL
23	IO_42	I/O	LVTTL/TTL	57	IO_43	I/O	LVTTL/TTL
24	IO_44	I/O	LVTTL/TTL	58	IO_45	I/O	LVTTL/TTL
25	IO_46	I/O	LVTTL/TTL	59	IO_47	I/O	LVTTL/TTL
26	GND			60	GND		
27	IO_48	I/O	LVTTL/TTL	61	IO_49	I/O	LVTTL/TTL
28	IO_50	I/O	LVTTL/TTL	62	IO_51	I/O	LVTTL/TTL
29	IO_52	I/O	LVTTL/TTL	63	IO_53	I/O	LVTTL/TTL
30	IO_54	I/O	LVTTL/TTL	64	IO_55	I/O	LVTTL/TTL
31	IO_56	I/O	LVTTL/TTL	65	IO_57	I/O	LVTTL/TTL
32	IO_58	I/O	LVTTL/TTL	66	IO_59	I/O	LVTTL/TTL
33	IO_60	I/O	LVTTL/TTL	67	IO_61	I/O	LVTTL/TTL
34	IO_62	I/O	LVTTL/TTL	68	IO_63	I/O	LVTTL/TTL

Table 13-6 : Front I/O Connector Pin Assignment TPMC644-10R

Note that the following pins are typically twisted pairs in the I/O cable: 1 & 35, 2 & 36, ..., 34 & 68.

Each single-ended I/O signal has an on-board pull-up to a common on-board reference. The reference is configurable by jumper (+3.3V, +5V or GND).

13.5.3 Front I/O Connector Pin Assignment TPMC644-11R/-13R

Pin	Signal	Dir	Level	Pin	Signal	Dir	Level
1	IO_0-	I/O	RS485/M-LVDS	35	IO_0+	I/O	RS485/M-LVDS
2	IO_1-	I/O	RS485/M-LVDS	36	IO_1+	I/O	RS485/M-LVDS
3	IO_2-	I/O	RS485/M-LVDS	37	IO_2+	I/O	RS485/M-LVDS
4	IO_3-	I/O	RS485/M-LVDS	38	IO_3+	I/O	RS485/M-LVDS
5	IO_4-	I/O	RS485/M-LVDS	39	IO_4+	I/O	RS485/M-LVDS
6	IO_5-	I/O	RS485/M-LVDS	40	IO_5+	I/O	RS485/M-LVDS
7	IO_6-	I/O	RS485/M-LVDS	41	IO_6+	I/O	RS485/M-LVDS
8	IO_7-	I/O	RS485/M-LVDS	42	IO_7+	I/O	RS485/M-LVDS
9	GND			43	GND		
10	IO_8-	I/O	RS485/M-LVDS	44	IO_8+	I/O	RS485/M-LVDS
11	IO_9-	I/O	RS485/M-LVDS	45	IO_9+	I/O	RS485/M-LVDS
12	IO_10-	I/O	RS485/M-LVDS	46	IO_10+	I/O	RS485/M-LVDS
13	IO_11-	I/O	RS485/M-LVDS	47	IO_11+	I/O	RS485/M-LVDS
14	IO_12-	I/O	RS485/M-LVDS	48	IO_12+	I/O	RS485/M-LVDS
15	IO_13-	I/O	RS485/M-LVDS	49	IO_13+	I/O	RS485/M-LVDS
16	IO_14-	I/O	RS485/M-LVDS	50	IO_14+	I/O	RS485/M-LVDS
17	IO_15-	I/O	RS485/M-LVDS	51	IO_15+	I/O	RS485/M-LVDS
18	IO_16-	I/O	RS485/M-LVDS	52	IO_16+	I/O	RS485/M-LVDS
19	IO_17-	I/O	RS485/M-LVDS	53	IO_17+	I/O	RS485/M-LVDS
20	IO_18-	I/O	RS485/M-LVDS	54	IO_18+	I/O	RS485/M-LVDS
21	IO_19-	I/O	RS485/M-LVDS	55	IO_19+	I/O	RS485/M-LVDS
22	IO_20-	I/O	RS485/M-LVDS	56	IO_20+	I/O	RS485/M-LVDS
23	IO_21-	I/O	RS485/M-LVDS	57	IO_21+	I/O	RS485/M-LVDS
24	IO_22-	I/O	RS485/M-LVDS	58	IO_22+	I/O	RS485/M-LVDS
25	IO_23-	I/O	RS485/M-LVDS	59	IO_23+	I/O	RS485/M-LVDS
26	GND			60	GND		
27	IO_24-	I/O	RS485/M-LVDS	61	IO_24+	I/O	RS485/M-LVDS
28	IO_25-	I/O	RS485/M-LVDS	62	IO_25+	I/O	RS485/M-LVDS
29	IO_26-	I/O	RS485/M-LVDS	63	IO_26+	I/O	RS485/M-LVDS
30	IO_27-	I/O	RS485/M-LVDS	64	IO_27+	I/O	RS485/M-LVDS
31	IO_28-	I/O	RS485/M-LVDS	65	IO_28+	I/O	RS485/M-LVDS
32	IO_29-	I/O	RS485/M-LVDS	66	IO_29+	I/O	RS485/M-LVDS
33	IO_30-	I/O	RS485/M-LVDS	67	IO_30+	I/O	RS485/M-LVDS
34	IO_31-	I/O	RS485/M-LVDS	68	IO_31+	I/O	RS485/M-LVDS

Table 13-7 : Front I/O Connector Pin Assignment TPMC644-11R/-13R

Note that the following pins are typically twisted pairs in the I/O cable: 1 & 35, 2 & 36, ..., 34 & 68.

Each differential signal pair has an on-board termination resistor (120R for RS485, 100R for M-LVDS).

13.5.4 Front I/O Connector Pin Assignment TPMC644-12R/-14R

Pin	Signal	Dir	Level	Pin	Signal	Dir	Level
1	IO_0-	I/O	RS485/M-LVDS	35	IO_0+	I/O	RS485/M-LVDS
2	IO_1-	I/O	RS485/M-LVDS	36	IO_1+	I/O	RS485/M-LVDS
3	IO_2-	I/O	RS485/M-LVDS	37	IO_2+	I/O	RS485/M-LVDS
4	IO_3-	I/O	RS485/M-LVDS	38	IO_3+	I/O	RS485/M-LVDS
5	IO_4-	I/O	RS485/M-LVDS	39	IO_4+	I/O	RS485/M-LVDS
6	IO_5-	I/O	RS485/M-LVDS	40	IO_5+	I/O	RS485/M-LVDS
7	IO_6-	I/O	RS485/M-LVDS	41	IO_6+	I/O	RS485/M-LVDS
8	IO_7-	I/O	RS485/M-LVDS	42	IO_7+	I/O	RS485/M-LVDS
9	GND			43	GND		
10	IO_8-	I/O	RS485/M-LVDS	44	IO_8+	I/O	RS485/M-LVDS
11	IO_9-	I/O	RS485/M-LVDS	45	IO_9+	I/O	RS485/M-LVDS
12	IO_10-	I/O	RS485/M-LVDS	46	IO_10+	I/O	RS485/M-LVDS
13	IO_11-	I/O	RS485/M-LVDS	47	IO_11+	I/O	RS485/M-LVDS
14	IO_12-	I/O	RS485/M-LVDS	48	IO_12+	I/O	RS485/M-LVDS
15	IO_13-	I/O	RS485/M-LVDS	49	IO_13+	I/O	RS485/M-LVDS
16	IO_14-	I/O	RS485/M-LVDS	50	IO_14+	I/O	RS485/M-LVDS
17	IO_15-	I/O	RS485/M-LVDS	51	IO_15+	I/O	RS485/M-LVDS
18	IO_32	I/O	LVTTL/TTL	52	IO_33	I/O	LVTTL/TTL
19	IO_34	I/O	LVTTL/TTL	53	IO_35	I/O	LVTTL/TTL
20	IO_36	I/O	LVTTL/TTL	54	IO_37	I/O	LVTTL/TTL
21	IO_38	I/O	LVTTL/TTL	55	IO_39	I/O	LVTTL/TTL
22	IO_40	I/O	LVTTL/TTL	56	IO_41	I/O	LVTTL/TTL
23	IO_42	I/O	LVTTL/TTL	57	IO_43	I/O	LVTTL/TTL
24	IO_44	I/O	LVTTL/TTL	58	IO_45	I/O	LVTTL/TTL
25	IO_46	I/O	LVTTL/TTL	59	IO_47	I/O	LVTTL/TTL
26	GND			60	GND		
27	IO_48	I/O	LVTTL/TTL	61	IO_49	I/O	LVTTL/TTL
28	IO_50	I/O	LVTTL/TTL	62	IO_51	I/O	LVTTL/TTL
29	IO_52	I/O	LVTTL/TTL	63	IO_53	I/O	LVTTL/TTL
30	IO_54	I/O	LVTTL/TTL	64	IO_55	I/O	LVTTL/TTL
31	IO_56	I/O	LVTTL/TTL	65	IO_57	I/O	LVTTL/TTL
32	IO_58	I/O	LVTTL/TTL	66	IO_59	I/O	LVTTL/TTL
33	IO_60	I/O	LVTTL/TTL	67	IO_61	I/O	LVTTL/TTL
34	IO_62	I/O	LVTTL/TTL	68	IO_63	I/O	LVTTL/TTL

Table 13-8 : Front I/O Connector Pin Assignment TPMC644-12R/-14R

Note that the following pins are typically twisted pairs in the I/O cable: 1 & 35, 2 & 36, ..., 34 & 68.

Each differential signal pair has an on-board termination resistor (120R for RS485, 100R for M-LVDS). Each single-ended I/O signal has an on-board pull-up to a common on-board reference. The reference is configurable by jumper (+3.3V, +5V or GND).

13.6 Rear I/O Interface

Also see the *Rear I/O Interface Signals* section in the *User FPGA* chapter!

The TPMC644 features 32 + 2 P14 Rear I/O signals, directly connected to dedicated User FPGA I/O pins (without any additional circuitry).

The signal levels on the rear I/O interface directly reflect the I/O standards used for the various rear I/O pins in the User FPGA design. Depending on the configured User FPGA Rear I/O Bank Supply, available I/O standards are either LVCMOS33 (for a 3.3V bank supply) or LVCMOS25 and LVDS_25 (for a 2.5V bank supply).

For single-ended rear I/O, every User FPGA RIO signal represents a rear I/O interface signal.

For differential rear I/O, two corresponding User FPGA RIO signals are required per differential pair on the rear interface.

The TPMC644 does not provide any on-board termination resistors for the rear I/O signals/lines. For LVDS_25, User FPGA internal termination must be used.

CAUTION!

The Rear I/O signals are direct FPGA I/O pin signals without any additional on-board transceiver or protection!

When used with a 3.3V bank supply, applying external voltage levels > 3.6V to the rear I/O signals will damage the FPGA device!

When used with a 2.5V bank supply, applying external voltage levels > 2.625V to the rear I/O signals will damage the FPGA device!

See the Rear I/O connector pin assignment on the next page.

13.7 Rear I/O Connector

13.7.1 Connector Type

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

Table 13-9 : Rear I/O Connector Type

13.7.2 Pin Assignment

Pin	I/O Signal	Level		Pin	I/O Signal	Level
1	RIO_0/0+	LVC MOS LVDS(+)		2	RIO_1/0-	LVC MOS LVDS(-)
3	RIO_2/1+	"		4	RIO_3/1-	"
5	RIO_4/2+	"		6	RIO_5/2-	"
7	RIO_6/3+	"		8	RIO_7/3-	"
9	RIO_8/4+	"		10	RIO_9/4-	"
11	RIO_10/5+	"		12	RIO_11/5-	"
13	RIO_12/6+	"		14	RIO_13/6-	"
15	RIO_14/7+	"		16	RIO_15/7-	"
17	RIO_16/8+	"		18	RIO_17/8-	"
19	RIO_18/9+	"		20	RIO_19/9-	"
21	RIO_20/10+	"		22	RIO_21/10-	"
23	RIO_22/11+	"		24	RIO_23/11-	"
25	RIO_24/12+	"		26	RIO_25/12-	"
27	RIO_26/13+	"		28	RIO_27/13-	"
29	RIO_28/14+	"		30	RIO_29/14-	"
31	RIO_30/15+	"		32	RIO_31/15-	"
33	-	-		34	-	-
35	-	-		36	-	-
37	-	-		38	-	-
39	-	-		40	-	-
41	-	-		42	-	-
43	-	-		44	-	-
45	-	-		46	-	-
47	-	-		48	-	-
49	-	-		50	-	-
51	-	-		52	-	-
53	-	-		54	-	-
55	-	-		56	-	-
57	-	-		58	-	-
59	-	-		60	-	-
61	-	-		62	-	-
63	RIO_62/31+	LVC MOS LVDS(+)		64	RIO_63/31-	LVC MOS LVDS(-)

Table 13-10 : Rear I/O Connector Pin Assignment

14 Appendix A (TPMC644 BRD)

The TPMC644 comes with a Board Reference Design (BRD) for the Spartan-7 User FPGA, factory programmed to the on-board SPI Flash.

Please see the TPMC644 BRD User Manual at www.tews.com for more information on the implemented functions, PCI BARs and register map.

The Board Reference Design is available upon request. Please contact support@tews.com.

15 Appendix B (Port Map Example)

```

entity tpmc644_uad_brd is
  port (

    -- Local Bus
    LB_RST_n      : in    std_logic; -- Local Bus Reset
    LB_CLK        : in    std_logic; -- Local Bus Clock (33 MHz)

    LB_CYC_n     : in    std_logic; -- Local Bus Cycle
    LB_R_W_n     : in    std_logic; -- Local Bus Read/Write# Access
    LB_D_A_n     : in    std_logic; -- Local Bus Data/Address# Phase
    LB_C_BE_n    : in    std_logic_vector( 3 downto 0); -- Local Bus Control/Byte Enable
    LB_AD        : inout std_logic_vector(31 downto 0); -- Local Bus Address/Data Bus
    --
    LB_MRDY_n    : in    std_logic; -- Local Bus Master Ready
    LB_MABT_n    : in    std_logic; -- Local Bus Master Abort
    LB_MLAST_n   : in    std_logic; -- Local Bus Master Last
    --
    LB_TRDY_n    : out   std_logic; -- Local Bus Target Ready
    LB_TTERM_n   : out   std_logic; -- Local Bus Target Termination
    LB_TERR_n    : out   std_logic; -- Local Bus Target Error
    --
    LB_INT_n     : out   std_logic; -- Local Bus Interrupt

    -- Clock
    -- EMCCLK      : in    std_logic; -- External Configuration Clock (53.2 MHz)

    USR_CLK      : in    std_logic; -- User Clock (40 MHz)
    BRG_CLK      : in    std_logic; -- Baudrate Generator Clock (7.3728 MHz)

    -- Digital IO
    FIO          : inout std_logic_vector(63 downto 0); -- Front-IO
    FIO_OE       : out   std_logic_vector(63 downto 0); -- Front-IO Output Enable

    RIO_VCCO_SEL : out   std_logic; -- Rear-IO Bank Voltage Selection

    RIO          : inout std_logic_vector(31 downto 0); -- Rear-IO
    RIO_EXT      : inout std_logic_vector(63 downto 62); -- Rear-IO Extended

    -- Misc
    BLD_OPT      : in    std_logic_vector( 2 downto 0); -- Build Option (Hardware Configuration)

    USR_SCL      : inout std_logic; -- User I2C Bus (EEPROM)
    USR_SDA      : inout std_logic; -- User I2C Bus (EEPROM)
    USR_LED      : out   std_logic_vector( 1 downto 0) -- User LEDs

  );
end tpmc644_uad_brd;

```

16 Appendix C (Design Constraints Example)

```

## #####
## General
## #####

# General Config Settings
set_property CFGBVS VCCO [current_design]
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property BITSTREAM.GENERAL.COMPRESS FALSE [current_design]

# Use External Configuration Clock
set_property BITSTREAM.CONFIG.EXTMASTERCLK_EN div-2 [current_design]

# SPI Configuration Mode
set_property BITSTREAM.CONFIG.SPI_FALL_EDGE YES [current_design]
set_property CONFIG_MODE SPIx1 [current_design]
set_property BITSTREAM.CONFIG.SPI_BUSWIDTH 1 [current_design]

## #####
## Bank #13
## #####

# 2.5V or 3.3V - 3.3V is used as default

# Bank is not supported

## #####
## Bank #14
## #####

set_property -dict { PACKAGE_PIN N15 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[42]} ]
#set_property -dict { PACKAGE_PIN M21 IOSTANDARD LVCMOS33 } [ get_ports {CFG_D0_MOSI} ]
#set_property -dict { PACKAGE_PIN M22 IOSTANDARD LVCMOS33 } [ get_ports {CFG_D1_DIN} ]

#set_property -dict { PACKAGE_PIN N19 IOSTANDARD LVCMOS33 } [ get_ports {EMCLK} ]
#set_property -dict { PACKAGE_PIN N17 IOSTANDARD LVCMOS33 } [ get_ports {CFG_FCS_B} ]

set_property -dict { PACKAGE_PIN N18 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[43]} ]
set_property -dict { PACKAGE_PIN T21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[44]} ]
set_property -dict { PACKAGE_PIN T22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[45]} ]
set_property -dict { PACKAGE_PIN V21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[46]} ]
set_property -dict { PACKAGE_PIN W22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[47]} ]
set_property -dict { PACKAGE_PIN U22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[48]} ]
set_property -dict { PACKAGE_PIN V22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[49]} ]
set_property -dict { PACKAGE_PIN W21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[50]} ]
set_property -dict { PACKAGE_PIN Y21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[51]} ]
set_property -dict { PACKAGE_PIN U20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[52]} ]
set_property -dict { PACKAGE_PIN V20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[53]} ]
set_property -dict { PACKAGE_PIN U18 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[54]} ]
set_property -dict { PACKAGE_PIN U19 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[55]} ]

set_property -dict { PACKAGE_PIN P15 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {USR_SCL} ]
set_property -dict { PACKAGE_PIN P16 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {USR_SDA} ]

set_property -dict { PACKAGE_PIN P17 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {USR_LED[0]} ]
set_property -dict { PACKAGE_PIN R18 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {USR_LED[1]} ]

set_property -dict { PACKAGE_PIN R19 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[63]} ]
set_property -dict { PACKAGE_PIN R20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[63]} ]
set_property -dict { PACKAGE_PIN T19 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[62]} ]
set_property -dict { PACKAGE_PIN T20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[62]} ]
set_property -dict { PACKAGE_PIN Y20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[61]} ]
set_property -dict { PACKAGE_PIN AA20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[61]} ]
set_property -dict { PACKAGE_PIN Y22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[60]} ]
set_property -dict { PACKAGE_PIN AA22 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[60]} ]
set_property -dict { PACKAGE_PIN W18 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[59]} ]
set_property -dict { PACKAGE_PIN Y19 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[59]} ]
set_property -dict { PACKAGE_PIN AA21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[58]} ]
set_property -dict { PACKAGE_PIN AB21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[58]} ]
set_property -dict { PACKAGE_PIN AB19 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[57]} ]
set_property -dict { PACKAGE_PIN AB20 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[56]} ]

set_property -dict { PACKAGE_PIN V18 IOSTANDARD LVCMOS33 } [ get_ports {BLD_OPT[2]} ]
set_property -dict { PACKAGE_PIN V19 IOSTANDARD LVCMOS33 } [ get_ports {BLD_OPT[1]} ]
set_property -dict { PACKAGE_PIN T16 IOSTANDARD LVCMOS33 } [ get_ports {BLD_OPT[0]} ]

## #####
## Bank #15
## #####

# 2.5V or 3.3V - 3.3V is used as default

set_property -dict { PACKAGE_PIN A21 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[42]} ]

```



```

set_property -dict { PACKAGE_PIN K6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[2]} ]
set_property -dict { PACKAGE_PIN J6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[2]} ]
set_property -dict { PACKAGE_PIN J8 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[3]} ]
set_property -dict { PACKAGE_PIN J7 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[3]} ]
set_property -dict { PACKAGE_PIN L7 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[4]} ]
set_property -dict { PACKAGE_PIN L6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[4]} ]
set_property -dict { PACKAGE_PIN L5 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[5]} ]
set_property -dict { PACKAGE_PIN K5 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[5]} ]
set_property -dict { PACKAGE_PIN L4 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[6]} ]
set_property -dict { PACKAGE_PIN K4 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[6]} ]
set_property -dict { PACKAGE_PIN L8 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[7]} ]
set_property -dict { PACKAGE_PIN K8 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[7]} ]

set_property -dict { PACKAGE_PIN M8 IOSTANDARD LVCMOS33 } [ get_ports {USR_CLK} ]
set_property -dict { PACKAGE_PIN M4 IOSTANDARD LVCMOS33 } [ get_ports {BRG_CLK} ]

set_property -dict { PACKAGE_PIN M3 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[8]} ]

set_property -dict { PACKAGE_PIN L1 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[29]} ]
set_property -dict { PACKAGE_PIN K1 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[28]} ]
set_property -dict { PACKAGE_PIN M2 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[27]} ]
set_property -dict { PACKAGE_PIN M1 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[26]} ]

set_property -dict { PACKAGE_PIN N4 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[8]} ]
set_property -dict { PACKAGE_PIN N3 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[9]} ]

set_property -dict { PACKAGE_PIN P1 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[25]} ]
set_property -dict { PACKAGE_PIN N1 IOSTANDARD LVCMOS33 DRIVE 12 SLEW SLOW } [ get_ports {LB_AD[24]} ]

set_property -dict { PACKAGE_PIN P3 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[9]} ]
set_property -dict { PACKAGE_PIN P2 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[10]} ]
set_property -dict { PACKAGE_PIN P8 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[10]} ]
set_property -dict { PACKAGE_PIN N8 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[11]} ]
set_property -dict { PACKAGE_PIN N5 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[11]} ]
set_property -dict { PACKAGE_PIN M5 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[12]} ]
set_property -dict { PACKAGE_PIN P6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[12]} ]
set_property -dict { PACKAGE_PIN N6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[13]} ]
set_property -dict { PACKAGE_PIN P7 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[13]} ]
set_property -dict { PACKAGE_PIN N7 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[14]} ]
set_property -dict { PACKAGE_PIN R7 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[14]} ]
set_property -dict { PACKAGE_PIN R6 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[15]} ]
set_property -dict { PACKAGE_PIN R5 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[15]} ]
set_property -dict { PACKAGE_PIN R4 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO[16]} ]
set_property -dict { PACKAGE_PIN R3 IOSTANDARD LVCMOS33 DRIVE 8 SLEW SLOW } [ get_ports {FIO_OE[16]} ]

```

```

## #####
## Bank #36

# Bank is not supported

## #####

```