

The Embedded I/O Company



TAMC640

Virtex-5 AMC with FMC Slot

Version 1.0

User Manual

Issue 1.0.4

January 2012

TEWS TECHNOLOGIES GmbH

Am Bahnhof 7 25469 Halstenbek, Germany

Phone: +49 (0) 4101 4058 0 Fax: +49 (0) 4101 4058 19

e-mail: info@tews.com www.tews.com

TAMC640-10R

XC5VLX50T-1, 256 MB DDR2, 2MB QDR-II,
Mid-Size front panel

TAMC640-11R

same as TAMC640-10R but Full-Size front panel

TAMC640-12R

XC5VLX85T-1, 256 MB DDR2, 2MB QDR-II,
Mid-Size front panel

TAMC640-13R

same as TAMC640-12R but Full-Size front panel

TAMC640-14R

XC5VSX50T-1, 256 MB DDR2, 2MB QDR-II Mid-
Size front panel

TAMC640-15R

same as TAMC640-14R but Full-Size front panel

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

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

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

Style Conventions

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

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

Access terms are described as:

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

©2011-2012 by TEWS TECHNOLOGIES GmbH

All trademarks mentioned are property of their respective owners.

Issue	Description	Date
1.0.0	Initial Issue	May 2011
1.0.1	corrected value of "Power Requirements as per Module Current Requirement Record (max.)" in chapter "Technical Specification"	June 2011
1.0.2	<ul style="list-style-type: none"> - Revised chapters "SPI-Flash" and "FPGA Configuration". Added information and warning about the relationship between indirect SPI-programming with Xilinx Impact and the FPGA Mode Pins. - Corrected SPI Interface Pinning in Appendix B 	July 2011
1.0.3	Update to V1.0 Rev. B: <ul style="list-style-type: none"> - Changed default FPGA configuration mode to "Master SelectMap" - Changed DIP-Switch 3 from "USER_SWITCH_CPLD" to "FPGA Configuration Source selection" - Table with worst case FPGA Configuration Times added - Pin assignment for (post configuration) user SPI access changed - Chapter "Design Help" moved to the end of the document, and chapter "Troubleshooting" added. 	November 2011
1.0.4	Correction of the interchanged DIP-Switch SW1 and SW2 descriptions. This affects: <ul style="list-style-type: none"> - Table 5-12 "configuration DIP-Switch SW1-SW2 Settings" in chapter 5.11 "JTAG" - Table 6-1 "DIP Switch" in chapter 6.2 "DIP-Switch" Samtec Part Number of the FMC Connector corrected (chapter 9.4)	January 2012

Table of Contents

1	PRODUCT DESCRIPTION	8
2	TECHNICAL SPECIFICATION	10
3	HANDLING AND OPERATION INSTRUCTION	12
3.1	ESD Protection	12
3.2	Thermal Considerations	12
3.3	Mid-Size Option Usage Restrictions	12
3.4	I/O Signaling Voltages	12
3.5	Voltage Limits on FMCs	12
4	IPMI SUPPORT	13
4.1	Temperature and Voltage Sensors.....	13
4.1.1	Sensor Locations	13
4.2	FRU Information	14
4.2.1	Internal Use Area	14
4.2.2	Board Info Area.....	15
4.2.3	Product Info Area	15
4.2.4	Multi Record Area	16
4.2.4.1	Module Current Requirements	16
4.2.4.2	AMC Point-to-Point Connectivity	16
4.2.4.3	Clock Configuration	17
4.2.5	Modifying FRU Records.....	17
5	FUNCTIONAL DESCRIPTION	18
5.1	AMC Interface	20
5.2	FMC Interface	20
5.2.1	VADJ.....	22
5.3	Memory Interfaces	22
5.3.1	SPI-Flash	22
5.3.2	QDR-II SRAM	23
5.3.3	DDR2 SDRAM	23
5.4	Reset	24
5.5	GPIO	25
5.6	I ² C	25
5.7	UART	26
5.8	Multi-Gigabit Transceiver (GTPs).....	27
5.9	Configuration.....	28
5.9.1	Board Configuration CPLD (BCC)	28
5.9.2	FPGA Configuration.....	29
5.9.3	Clock Configuration	31
5.10	Clocks	33
5.10.1	GTP Reference Clock Generator.....	34
5.11	JTAG.....	35
5.12	Thermal Management	36
6	BOARD CONFIGURATION	37
6.1	Overview	37
6.2	DIP-Switch	37
6.3	Battery	38
6.4	Debug Connector	38

7	INSTALLATION	39
7.1	Installation of a FMC Module	39
7.1.1	Using FMCs with Mid-Size faceplates	39
7.1.2	Voltage Limits on FMC Modules	39
7.2	AMC Module Insertion & Hot-Swap	40
7.2.1	Insertion	40
7.2.2	Extraction	40
8	INDICATORS	41
8.1	Front Panel LEDs	41
8.2	On board LEDs	41
9	I/O CONNECTORS	43
9.1	Overview	43
9.2	I/O Circuitry	43
9.2.1	Differential Signaling	43
9.3	AMC-Connector X4	44
9.4	FMC HPC Connector X2	46
9.5	Debug-Connector X1	49
10	DESIGN HELP	50
10.1	GTP Reference Clock Generator Configuration	50
10.2	Example Design	50
10.3	Troubleshooting	51
10.3.1	Board does not power up	51
10.3.2	DONE is always off	51
10.3.3	INIT LED stays illuminated (red)	51
11	APPENDIX A	52
12	APPENDIX B	58

List of Figures

FIGURE 1-1 : BLOCK DIAGRAM.....	9
FIGURE 4-1 : TEMP. SENSOR LOCATIONS.....	13
FIGURE 5-1 : TAMC640 FUNCTIONAL BLOCK DIAGRAM	18
FIGURE 5-2 : AMC INTERFACE DIAGRAM	20
FIGURE 5-3 : FMC INTERFACE TO VIRTEX-5 FPGA	21
FIGURE 5-4 : A FMC MODULE	22
FIGURE 5-5 : QDR-II SRAM INTERFACE TO FPGA.....	23
FIGURE 5-6 : DDR2 SDRAM INTERFACES TO FPGA	24
FIGURE 5-7 : RESET STRUCTURE	24
FIGURE 5-8 : I2C BUS STRUCTURE OVERVIEW.....	26
FIGURE 5-9 : GTP BLOCK DIAGRAM	27
FIGURE 5-10: BOARD CONFIGURATION CPLD BLOCK DIAGRAM.....	28
FIGURE 5-11: CLOCK CONFIGURATION	32
FIGURE 5-12: FPGA CLOCK SOURCES.....	33
FIGURE 5-13: JTAG-CHAIN SEGMENTATION	35
FIGURE 7-1 : PRE-INSERTION BOARD CONFIGURATION OVERVIEW.....	37
FIGURE 8-1 : USING FMCS WITH MID-SIZE FACEPLATES	39
FIGURE 9-1 : FRONT PANEL LED VIEW	41
FIGURE 9-2 : ON BOARD LED VIEW	41
FIGURE 10-1: CONNECTOR POSITIONS – SIDE 1	43
FIGURE 10-2: CONNECTOR POSITION – SIDE 2.....	43

List of Tables

TABLE 2-1 : TECHNICAL SPECIFICATION.....	11
TABLE 4-1 : TEMPERATURE AND VOLTAGE SENSORS	13
TABLE 4-2 : FRU INFORMATION	14
TABLE 4-3 : INTERNAL USE AREA.....	14
TABLE 4-4 : BOARD INFO AREA.....	15
TABLE 4-5 : PRODUCT INFO AREA.....	15
TABLE 4-6 : MODULE CURRENT REQUIREMENT (W/O FMC).....	16
TABLE 4-7 : AMC POINT-TO-POINT CONNECTIVITY	17
TABLE 4-8 : CLOCK CONFIGURATION	17
TABLE 5-1 : TAMC640 FPGA FEATURE OVERVIEW	19
TABLE 5-2 : FPGA BANK USAGE.....	19
TABLE 5-3: FMC-SUPPLIES	21
TABLE 5-4 : GENERAL PURPOSE I/O	25
TABLE 5-5 : I2C BUS SIGNALS	25
TABLE 5-6 : FPGA UART	26
TABLE 5-7 : WORST CASE FPGA CONFIGURATION TIMES	30
TABLE 5-8 : TCLK TRANSCEIVER CONFIGURATION.....	31
TABLE 5-9 : JITTER ATTENUATOR CONFIGURATION.....	31
TABLE 5-10: AVAILABLE FPGA CLOCKS.....	34
TABLE 5-11: PROGRAMMABLE GTP REFERENCE CLOCK.....	35
TABLE 5-12: CONFIGURATION DIP-SWITCH SW1-SW2 SETTINGS	36
TABLE 6-1 : DIP-SWITCH.....	37
TABLE 7-1 : VOLTAGE LIMITS ON FMC MODULES	39
TABLE 7-2 : HOT-SWAP STATES	40
TABLE 8-1 : FRONT PANEL LEDS	41
TABLE 8-2 : BOARD-STATUS LEDS	42
TABLE 9-1: PIN ASSIGNMENT AMC CONNECTOR X4	45
TABLE 9-2: PIN ASSIGNMENT FMC-CONNECTOR X2 ROW F-K.....	47
TABLE 9-3: PIN ASSIGNMENT FMC-CONNECTOR X2 ROW A-E	48
TABLE 9-4: PIN ASSIGNMENT DEBUG CONNECTOR X1	49

1 Product Description

The TAMC640 is a standard single Mid-Size or Full-Size AMC module providing a user configurable Virtex-5 FPGA. The integrated PCIe Endpoint Block of the Virtex-5 can be used to build an x1, x4 or x8 PCIe link via AMC Port 4-11. The implementation of other protocols like SRIO or XAUI is also possible. AMC Ports 0 & 1, commonly used for Gigabit Ethernet, are also connected to the FPGA. The integrated Gigabit Ethernet MACs of the Virtex-5 allow fast and easy protocol implementation.

To allow direct board-to-board communication, AMC Ports 12-17 are connected to Virtex-5 I/Os, allowing AC-coupled LVDS communication with a port speed up to 1.0Gb/sec.

For flexible I/O solutions the TAMC640 provides a VITA 57.1 high pin count FMC Module slot, allowing active and passive signal conditioning. All FMC I/O lines are directly connected to the FPGA, which maintains the flexibility of the Select I/O technology of the Virtex-5 FPGA.

In addition, the FPGA is connected to the following external memories:

- two banks of DDR2 DRAM (up to 128 M x 16 (256 MB) each)
- one bank of QDR-II SRAM (up to 4 M x 18 (8 MB))

Multiple clocks from the AMC-interface, the FMC and from on-board sources are supplied to the FPGA.

The FPGA is configured by a flash device, which is in-system programmable and able to store multiple code versions.

The TAMC640 supports encrypted FPGA bitstream usage. Encrypted FPGA bitstreams cannot be copied or reverse engineered, securing your intellectual property.

The IPMI Connectivity Records located inside the Module Management Controller (MMC) can be modified by the customer (e.g. via IPMI), to adapt to the different possible communication protocols (PCIe, SRIO, XAUI, ...).

User applications for the TAMC640 require the full ISE Foundation software, which must be purchased from Xilinx.

The Engineering Documentation TAMC640-ED includes all information needed for customer specific FPGA programming. The FPGA Development Kit TAMC640-FDK includes the engineering documentation, ucf-files with all necessary pin assignments and basic timing constraints, and a well documented VHDL example application. This example application is called TPLD002 (Tews Programmable Logic Design) and covers the main functionalities of the TAMC640 like DMA capable PCIe endpoint with interrupt support, register mapping, DDR2 and QDR-II memory access and basic I/O to the FMC slot. It comes as a Xilinx ISE project with source code and as a ready-to-download bitstream. It is the basis for fast and reliable customer application development, and can significantly reduce time to market.

Software support for the TPLD002 is available for all major operating systems.

In-circuit programming and debugging of the FPGA design (e.g. using Xilinx "ChipScope") is supported. The Program and Debug Box TA900 allows access to the module while it is inserted in a system. It provides access to the module's JTAG Chain, the UART of the on-board Module Management Controller (MMC) and to two user pins of the Virtex-5 FPGA. If a UART core is implemented in the FPGA, serial communication via the TA900 is possible.

The TA900 can be accessed by USB 2.0 and by a 14-pin JTAG Header (e.g. for connecting a Xilinx Platform Cable).

For First-Time-Buyers the TA900 and the TAMC640-ED or TAMC640-FDK is recommended.

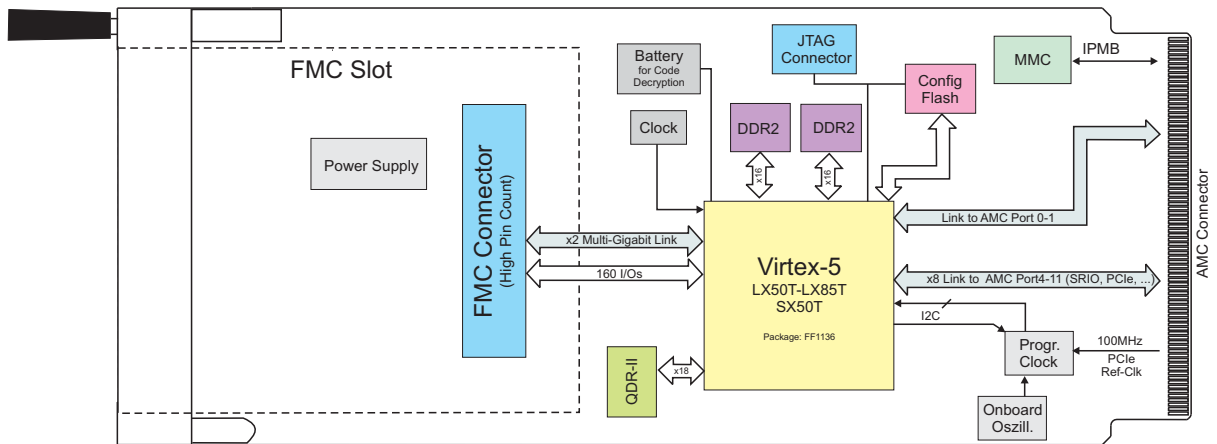


Figure 1-1 : Block Diagram

2 Technical Specification

AMC Interface	
Mechanical Interface	Advanced Mezzanine Card (AMC) Interface conforming to PICMG® AMC.0 R2.0 (Advanced Mezzanine Card Base Specification) Module Type: Single Mid-Size module or Single Full-Size module (see data sheet for order information)
Electrical Interface	Virtex-5 GTPs connected to AMC port 0, 1, 4 - 11 Virtex-5 LVDS Transceivers connected to port 12-15 & 17 TCLKA-D connected to FPGA
IPMI	
IPMI Version	1.5
Front Panel LEDs	Blue Hot-Swap LED Red Failure Indication LED (LED1) Green Board OK / User LED (LED2)
On Board Devices	
PCIe Endpoint	Virtex-5 integrated PCI Express Endpoint Block
User configurable FPGA	Virtex-5 (Xilinx) XC5VLX50T, XC5VLX85T, XC5VSX50T (see data sheet for order information)
Configuration Flash	2 x Xilinx XCF32P, 32 MBit each
SPI-Flash	MP25P64 (Micron) 64 Mbit (can be used for FPGA configuration)
DDR2 SDRAM	2 x Micron MT47H64M16 (64M x 16)
QDR-II SRAM	1 x IDT 71P74804 (1M x 18)
Programmable Clock Generator	Si5338B (Silicon Labs)
I/O Interface	
I/O Connector	FMC high pin count slot according to VITA 57.1 (FPGA Mezzanine Card (FMC) Standard)
User Defined Signals	80 differential or 160 single-ended I/O plus 4 differential Clocks
Multi-Gigabit-Interfaces	2 gigabit data plus 1 gigabit reference clocks

Physical Data		
Power Requirements	Depends on FPGA design. 400 mA typical @ +12V DC (Payload Power, Blank FPGA) 40 mA typical @ +3.3V DC (Management Power) 3.3A as per Module Current Requirement Record (max.) Additional power is used by the FMC.	
Temperature Range	Operating	0°C to +70°C
	Storage	-40°C to +85°C
MTBF	306000 h MTBF values shown are based on calculation according to MIL-HDBK-217F and MIL-HDBK-217F Notice 2; Environment: G _B 20°C. The MTBF calculation is based on component FIT rates provided by the component suppliers. If FIT rates are not available, MIL-HDBK-217F and MIL-HDBK-217F Notice 2 formulas are used for FIT rate calculation.	
Humidity	5 – 95 % non-condensing	
Weight	190 g	

Table 2-1 : Technical Specification

3 Handling and Operation Instruction

3.1 ESD Protection



The TAMC640 is sensitive to static electricity. Packing, unpacking and all other handling of the TAMC640 has to be done in an ESD/EOS protected Area.

3.2 Thermal Considerations



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

3.3 Mid-Size Option Usage Restrictions



Please note that the Mid-Size module has restrictions to its usage because of a component height violation. It is within the responsibility of the user to carefully check if the Mid-Size module with its component height violation can be used in the system. Otherwise damage to the TAMC640 or the slot it is used in may occur!
Refer to the chapter “Using FMCs with Mid-Size faceplates” for details.

3.4 I/O Signaling Voltages



The FPGA I/O-Lines to the FMC Slot are directly connected to the FPGA I/O pins. The I/O voltage of these FPGA I/O pins is 3.3V maximum.

The FPGA I/O pins are NOT 5V tolerant.

3.5 Voltage Limits on FMCs



The AMC.0 specification limits the voltages on AMC modules. These limits also apply to mounted FMCs.
Refer to the chapter “Voltage Limits on FMC Modules” for details.

4 IPMI Support

The TAMC640 provides a Module Management Controller (MMC) that performs health monitoring, hot-swap functionality and Field Replaceable Unit (FRU) information storage. The MMC communicates via an Intelligent Platform Management Interface (IPMI) with its superordinated IPMI controller / shelf manager.

4.1 Temperature and Voltage Sensors

The MMC monitors on board sensors and reports sensor events to the superordinated IPMI controller / shelf manager.

Sensor Number	Signal Type	Thresholds ¹	Signal Monitored
0	Event	-	Hot-swap switch
1	Temperature	lcr Inc unc ucr	FPGA Temp.
2	Temperature	lcr Inc unc ucr	Board Temp.
3	Temperature	lcr Inc unc ucr	FMC Air Temp.
4	Voltage	lcr Inc unc ucr	PWR
5	Voltage	lcr Inc unc ucr	+12V (FMC)
6	Voltage	lcr Inc unc ucr	+5V
7	Voltage	lcr Inc unc ucr	V _{ADJ} (FMC)

Table 4-1 : Temperature and Voltage Sensors

4.1.1 Sensor Locations

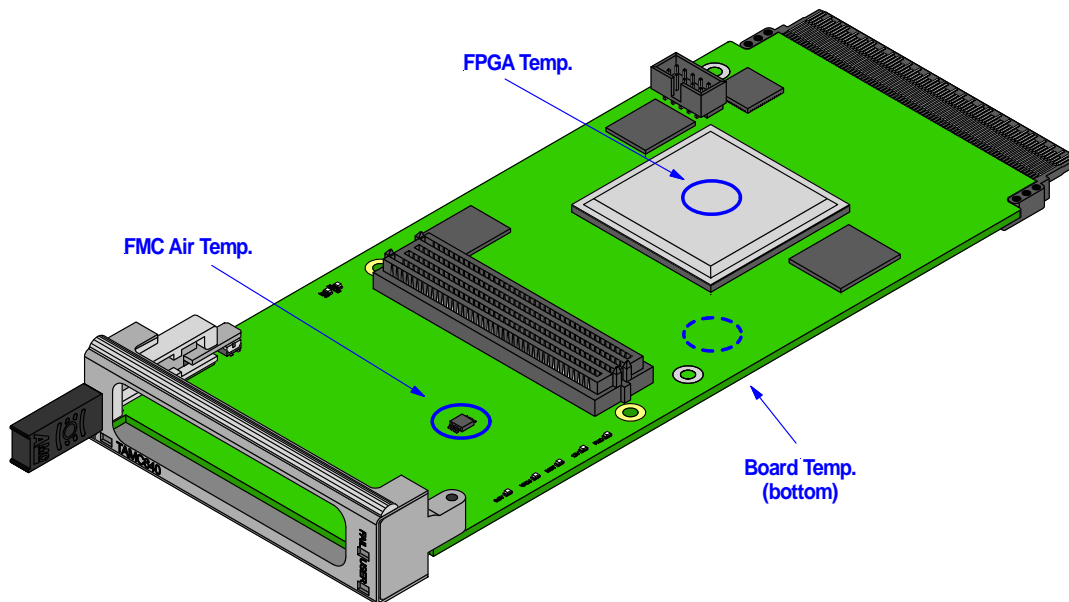


Figure 4-1 : Temp. Sensor Locations

¹ unr: upper non-recoverable, ucr: upper critical, unc: upper non-critical Inc: lower non-critical, lcr: lower critical, lnr: lower non-recoverable

4.2 FRU Information

The MMC stores the module FRU information in a non-volatile EEPROM. Some of the records are writeable to allow adapting the TAMC640 to user FPGA designs. If records are modified, the user is responsible to set the affected checksums to correct values.

Area	Size (in Bytes)	Writeable
Common Header	8	no
Internal Use Area	72	yes
Chassis Info Area	0	no
Board Info Area	variable	no
Product Info Area	variable	no
Multi Record Area	variable	see below
Module Current Requirements	variable	yes
AMC Point-to-Point Connectivity	variable	yes
Clock Configuration	variable	yes

Table 4-2 : FRU Information

4.2.1 Internal Use Area

The TAMC640 uses the Internal Use Area to store default FMC-slot settings for the case that a module is present, but no valid FRU information is found. The value of “Fallback-Voltage for VADJ” determines what happens in this case. If the “Fallback-Voltage for VADJ” is set to 0x0000, the module won’t turn on. Any other value sets VADJ to “Fallback-Voltage for VADJ” * 10mV, as long this value is within the range defined by Minimum VADJ and Maximum VADJ. Example: 0x00FA = 250 * 10mV = 2.5V.

Product Information	Value
Internal Use Format Version	0x01
TEWS IUA Format-Version	0x01
Present FMC Slots	0x01 – FMC-slot #0
Fallback-Voltage for VADJ	0x0000 – if no valid FMC-FRU is found, the TAMC640 won’t turn on.
Minimum VADJ	0x0078 – 1200mV for TAMC640
Maximum VADJ	0x014A – 3300mV for TAMC640

Table 4-3 : Internal Use Area

The whole Internal Use Area is writeable, but if changes become necessary, only the Fallback-Voltage for VADJ should be altered.

4.2.2 Board Info Area

Product Information	Value
Version	0x01
Language Code	0x00 - English
Manufacturer date/time	determined at manufacturing
Board manufacturer	TEWS TECHNOLOGIES GmbH
Board product name	TAMC640
Board serial number	determined at manufacturing (see board label)
Board part number	TAMC640-xxR -xx = -10 / -11 / -12 / -13 / -14 / -15

Table 4-4 : Board Info Area

4.2.3 Product Info Area

Product Information	Value
Version	0x01
Language Code	0x00 - English
Product manufacturer	TEWS TECHNOLOGIES GmbH
Product name	TAMC640
Board part/model number	TAMC640-xxR -xx = -10 / -11 / -12 / -13 / -14 / -15
Product version	V1.0 Rev. B (see board label)
Product serial number	determined at manufacturing (see board label)
Asset tag	= Product serial Number

Table 4-5 : Product Info Area

4.2.4 Multi Record Area

4.2.4.1 Module Current Requirements

The “Current Draw” value holds the Payload Power (PWR) requirement of the module given as current requirement in units of 0.1A at 12V.

The AMC module announces the sum of “Current Draw” and FMC Current Requirement as current demand to the shelf manager. If the power budget for the AMC slot is smaller than this value, the shelf manager may not enable Payload power for the slot.

If required, the “Current Draw” value in the Module Current Requirements record may be modified to a value that falls within the given power budget. Make sure that the modified value still satisfies the AMC module power requirements for the actual FPGA content.

Product Information	Value
Current Draw	0x2F (4.7 A)

Table 4-6 : Module Current Requirement (w/o FMC)

4.2.4.2 AMC Point-to-Point Connectivity

The TAMC640’s Virtex-5 FPGA allows implementing a wide range of interfaces (Serial RapidIO, PCI-Express, Gig.-Eth., XAUI, etc.). The MMC stores a Connectivity Record for each interface that is implemented by the TAMC640. By default, the MMC of the TAMC640 stores the following Connectivity Records:

- 2x GbE Links on AMC Ports 0-1
- x1 2.5 Gbps PCI-Express Gen1 Link on AMC Port 4
- x4 2.5 Gbps PCI-Express Gen1 Link on AMC Port 4-7
- x8 2.5 Gbps PCI-Express Gen1 Link on AMC Port 4-11

Channel	Port	Link Type	Link Type Extension	Link Grouping ID	Asymmetric Match
0	0	AMC.2 Ethernet	1000BASE-BX Ethernet Link	Single Channel Link	exact match
1	1	AMC.2 Ethernet	1000BASE-BX Ethernet Link	Single Channel Link	exact match
2	4	AMC.1 PCI Express	Gen 1 PCI Express, non-SSC	Single Channel Link	matches with 10
			Gen 1 PCI Express, SSC		
3	4-7	AMC.1 PCI Express	Gen 1 PCI Express, non-SSC	Single Channel Link	matches with 10
			Gen 1 PCI Express, SSC		
4	4-7	AMC.1 PCI Express	Gen 1 PCI Express, non-SSC	0x1	matches with 10
			Gen 1 PCI Express, SSC		

5	8-11	AMC.1 PCI Express	Gen 1 PCI Express, non-SSC	0x1	matches with 10
			Gen 1 PCI Express, SSC		

Table 4-7 : AMC Point-to-Point Connectivity

4.2.4.3 Clock Configuration

AMC FCLKA should be used as the PCI Express Reference Clock. TCLKA – D are all inputs by default, but can independently be changed to outputs.

Clock ID	Clock Features	Clock Family	Clock Accuracy	Clock Frequency
FCLKA	Clock Receiver, connected through Jitter Attenuator and programmable Clock Generator	PCI Express	PCI Express Gen 1	100 MHz nom
TCLKA	Clock Receiver	misc	-	-
TCLKB	Clock Receiver	misc	-	-
TCLKC	Clock Receiver	misc	-	-
TCLKD	Clock Receiver	misc	-	-

Table 4-8 : Clock Configuration

4.2.5 Modifying FRU Records

Some of the records are writeable to allow adapting the TAMC640 to user FPGA designs. If records are modified, the user is responsible to set the affected checksums to correct values.

5 Functional Description

This chapter gives a brief overview of the various module functions.

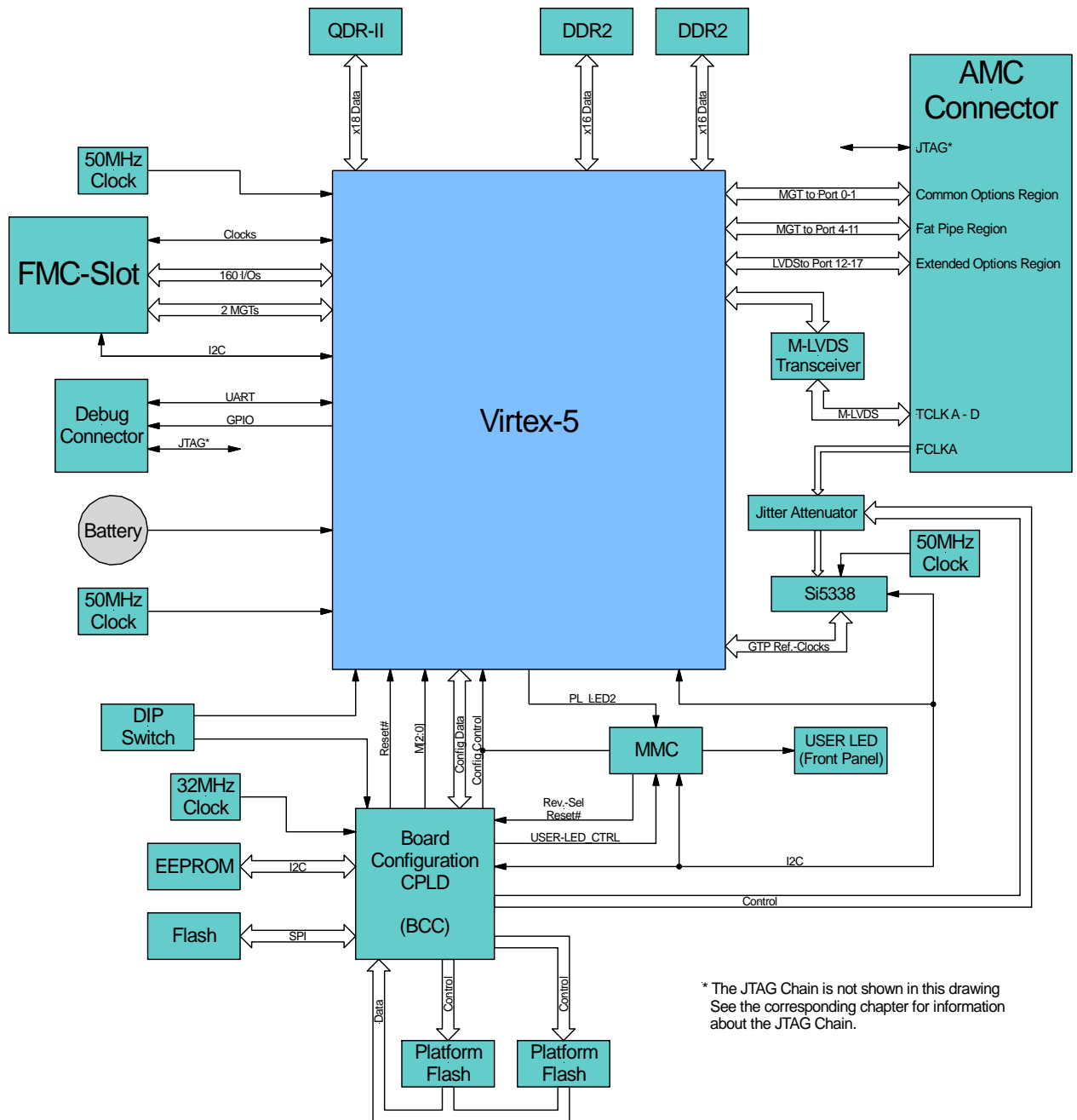


Figure 5-1 : TAMC640 Functional Block Diagram

The FPGA is a Virtex-5 LX50T, LX85T or SX50T FPGA. Each FPGA provides four Gigabit Ethernet MACs, and one Endpoint Blocks for PCI Express.

Virtex-5	Logic Cells	DSP48A1 Slices	Block RAM (Kb)	CMTs	Ethernet MACs	PCIe Endpoint	GTP / GTX Transceiver	Power PC
LX50T	46,080	48	2,160	6	4	1	12 / -	-
LX85T	82,944	48	3,888	6	4	1	12 / -	-
SX50T	52,224	288	4,752	6	4	1	12 / -	-

Table 5-1 : TAMC640 FPGA Feature Overview

The FPGA is equipped with 14 I/O banks and 12 Multi-Gigabit Transceivers.

I/O Bank	V _{CC0}	V _{REF}	Signals	Remarks
0	2.5V	-	Configuration	no user I/Os
1	VCC_B	VREF_B_M2C	FMC HB[17:21]	
2	2.5V	-	AMC Tx12 -15	+ Configuration
3	VADJ	VREF_A_M2C	FMC LA[17:23]	+ 50MHz Clock
4	2.5V	-	AMC TCLK FMC CLK	+ AMC Tx 17
11	VADJ	VREF_A_M2C	FMC LA[17:33]	
12	1.8V	0.9V	QDR-II Bank 0	
13	VADJ	VREF_A_M2C	FMC LA[00:16]	
15	VADJ	VREF_A_M2C	FMC HA[00:16]	
17	1.8V	0.9V	DDR2 Bank 1	
18	1.8V	0.9V	DDR2 Bank 0	+ AMC Rx 12-15 & 17
19	VCC_B	VREF_B_M2C	FMC HB[00:16]	
20	1.8V	-	QDR-II Bank 0	
21	1.8V	0.9V	DDR2 Bank 1	+ GPIO
22	1.8V	0.9V	DDR2 Bank 0	+ GPIO
GTP Bank	Description			Remarks
112 (X0Y3)	AMC Backplane, Port 6 & 7			
114 (X0Y2)	AMC Backplane, Port 8 & 9			
116 (X0Y4)	AMC Backplane, Port 4 & 5			
118 (X0Y1)	AMC Backplane, Port 10 & 11			
120 (X0Y5)	AMC Backplane, Port 0 & 1			
122 (X0Y0)	FMC, DP 0 & 1			

Table 5-2 : FPGA Bank Usage

All FMC I/O lines are directly connected to the FPGA-pins. Refer to the Xilinx UG190: *Virtex-5 FPGA User Guide* for SelectIO interface signal standards, slew rate control and current drive strength capabilities.

The board supports FPGA configuration via JTAG, from a Platform Flash or a SPI-Flash.

5.1 AMC Interface

AMC Ports 0, 1 & 4 – 11 are connected to Virtex-5 GTP transceivers. Ports 12-17 are connected to Virtex-5 LVDS Receivers / Transmitters. All Ports provide on board AC-coupling on the Rx and Tx lines.

FCLKA, commonly used as PCI-Express reference clock, is feed into a powerful clock distribution that allows using this clock on any FPGA GTP transceiver needed.

TCLKA – TCLKD are accessible by the Virtex-5 FPGA via M-LVDS transceivers that allow their use as inputs or outputs. All TCLKx inputs are lead on FPGA global clock buffers.

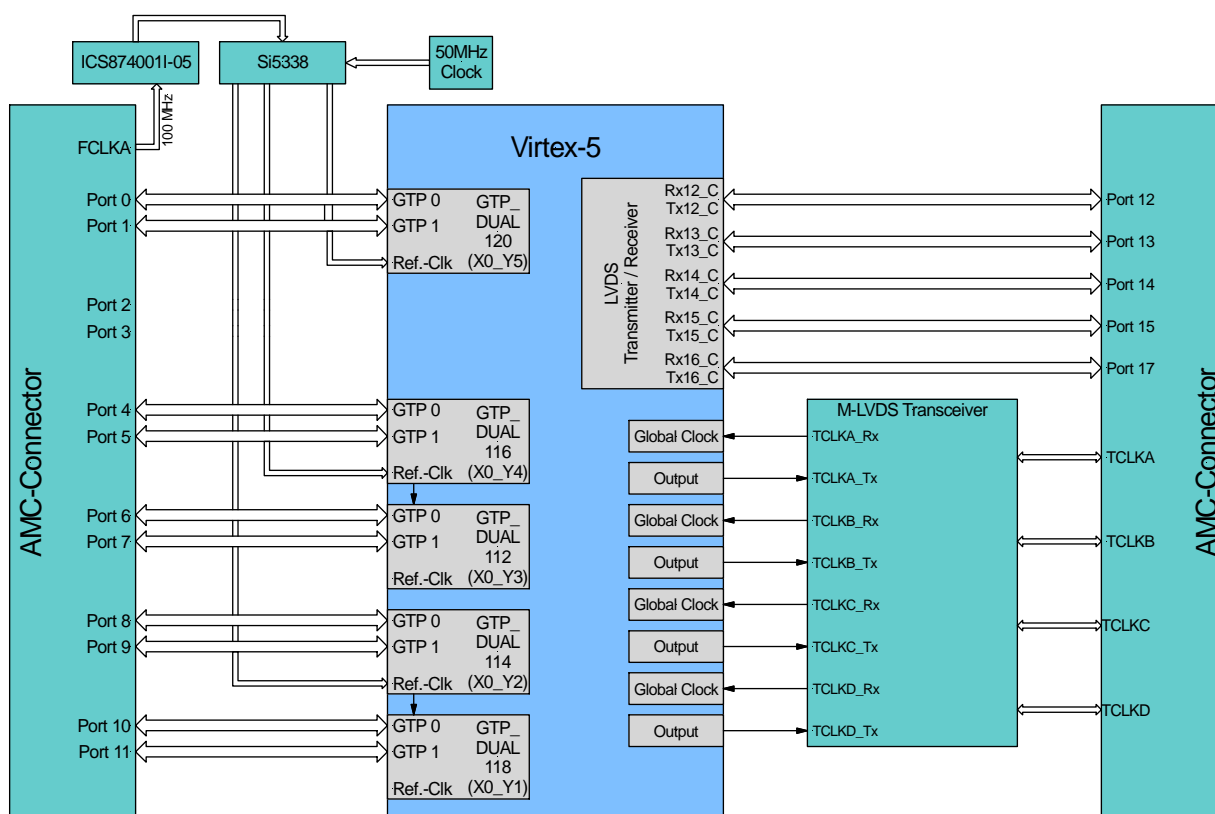


Figure 5-2 : AMC Interface Diagram

5.2 FMC Interface

Instead of a front I/O Connector, the TAMC640 offers a FPGA Mezzanine Card (FMC) module slot. This allows a wide range of connectors to be used with the TAMC640 and customer specific I/O solutions can be easily applied.

The FMC module can adapt the TAMC640 to various I/O standards, either mechanical (connector) or electrical.

The TAMC640 implements the High Pin Count (HPC) option of the VITA 57.1 specification. It offers almost the full set of connectivity options for the High Pin Count option:

- 160 single-ended or 80 differential user defined signals⁴ differential clocks
- 2 GTP links
- 1 GTP reference clock

The geographic address pins GA[0:1] are wired to a default of “00”. The FMC’s present signal (FMC_PRESENT_1V8#) and the FMC I²C bus are also connected to the FPGA.

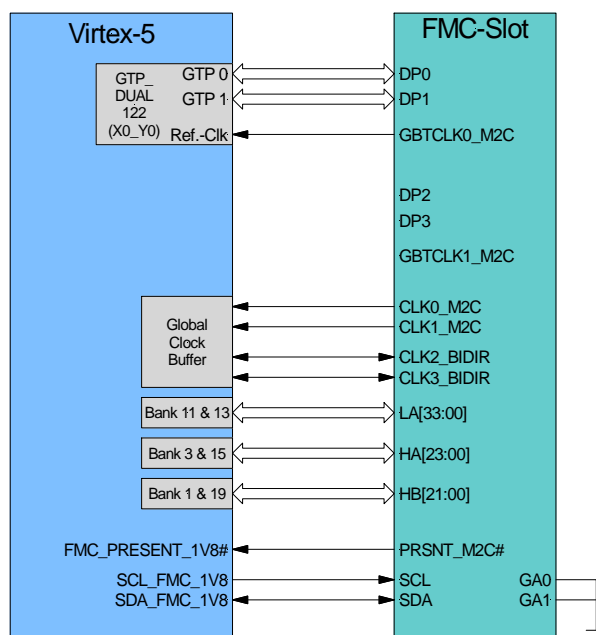


Figure 5-3 : FMC Interface to Virtex-5 FPGA

Refer to chapter “X2 FMC HPC Connector” for the FMC HPC Connector pin assignment.

The TAMC640 supports the maximum current for each FMC supply, as defined for a High Pin Count module slot.

Supply Voltage	Range	Max Amps
VADJ	1.2V ² - 3.3V	4 A
VIO_B_M2C	1.2V ³ - VADJ	1.15 A
VREF_A_M2C	0V - VADJ	1 mA
VREF_B_M2C	0V – VIO_B_M2C	1 mA
3.3VAUX	3.3V	20 mA
3.3V	3.3V	3 A
12V	12V	1 A

Table 5-3: FMC-Supplies

² Vita 57.1 defines this voltage as 0 -3.3V. On the TAMC640, this is limited to 1.2 – 3.3V, because the FPGA I/O buffer will not work with voltages below 1.2V.

³ Vita 57.1 defines this voltage as 0 -VADJ. On the TAMC640, this is limited to 1.2 –VADJ, because the FPGA I/O buffer will not work with voltages below 1.2V. If HB[21:00] are not used on the FMC, this voltage can be left unconnected.

The FMC standard is described in: VITA 57.1, available at www.vita.com/fmc.

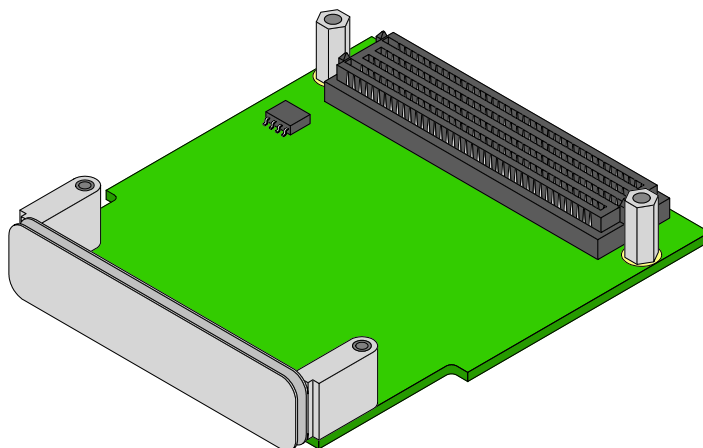


Figure 5-4 : A FMC Module

5.2.1 VADJ

If a FMC is present, the TAMC640 MMC reads the FRU information from the FMC’s I²C-EEPROM to determine how VADJ has to be adjusted. It uses the value in the “Nominal Voltage” field of the “DC Load” record for VADJ.

If a FMC is present, but no valid FRU information is found (because the FMC EEPROM is empty or does not exist), the TAMC640 MMC uses its “Fallback-Voltage for VADJ” setting that is stored in the Internal Use Area of the TAMC640 FRU information. To avoid damage to a plugged FMC, the “Fallback-Voltage for VADJ” is set to “0x0000” by default, which means that the module will not be turned on by the MMC.

If no FMC is present, the TAMC640 is turned on, with a VADJ set to 1.8V.

5.3 Memory Interfaces

The TAMC640 is equipped with two banks of 128 Mbytes, 16 bit wide DDR2 SDRAM, one bank of 2 Mbytes, 18 bit wide QDR-II SRAM and one 64-Mbit non-volatile SPI-Flash.

5.3.1 SPI-Flash

The TAMC640 provides a Numonyx M25P64 64-Mbit serial Flash memory, which can be used as FPGA configuration source. After configuration, it is accessible from the FPGA, so it also can be used for code or user data storage.



Before accessing the SPI-Flash (e.g. using Xilinx Impact), the FPGA Mode Pins must be set to “Master SPI” (factory default for V1.0 Rev. B).
If the FPGA Mode Pins are set to a slave configuration Mode (factory default of V1.0 Rev. A), damage to the device will occur during SPI programming via Impact.

See chapter “Board Configuration CPLD” for more details.

5.3.2 QDR-II SRAM

The TAMC640 provides a total of 2 MByte (18MBit) QDR-II SRAM per default (larger memories are possible).

The FPGA has access to one QDR-II SRAM device with 1 Mbit depth at 18 bit data bus width.

The TAMC640 uses Burst of 4 QDR-II SRAM to lower address-bus switching speed and simultaneously achieve read and write accesses to independent addresses of the SRAM without any wait cycles.

The maximum RAM clock frequency depends on FPGA speed and available routing resources.

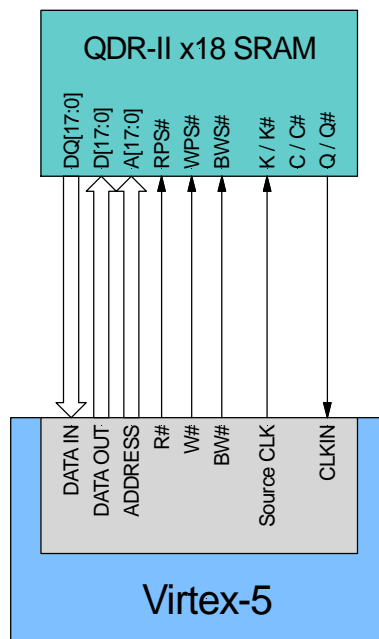


Figure 5-5 : QDR-II SRAM Interface to FPGA

5.3.3 DDR2 SDRAM

The TAMC640 provides two MT47H64M16 DDR2 memory components with 128 MByte DDR2 SDRAM at 16 Bit data bus width each.

Both DDR2 SDRAMs have fully independent interfaces to the FPGA.

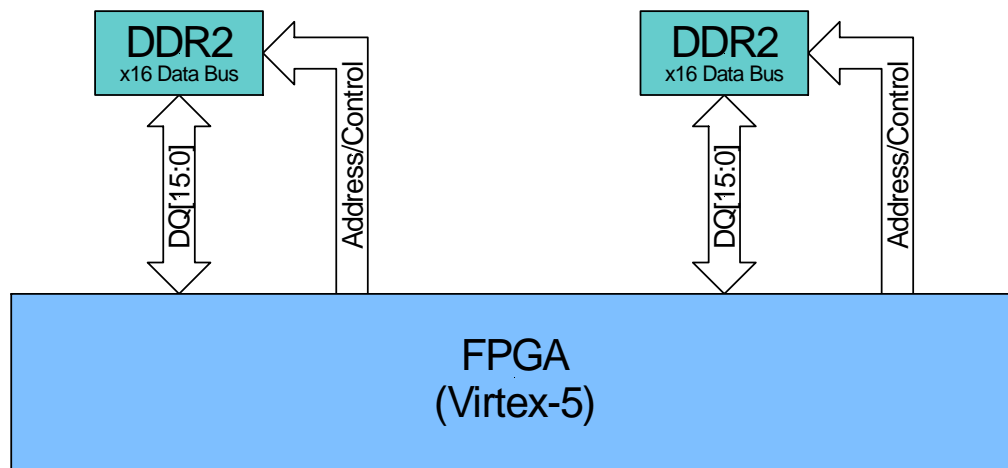


Figure 5-6 : DDR2 SDRAM Interfaces to FPGA

For details regarding the DDR2 SDRAM interface, please refer to the DDR2 SDRAM datasheet and the Xilinx UG086: *Xilinx Memory Interface Generator (MIG) User Guide*.

5.4 Reset

The MMC generates the reset signal to the TAMC640 payload devices. It is connected to the Board Configuration CPLD (BCC) that vice versa generates the reset signal for the FPGA.

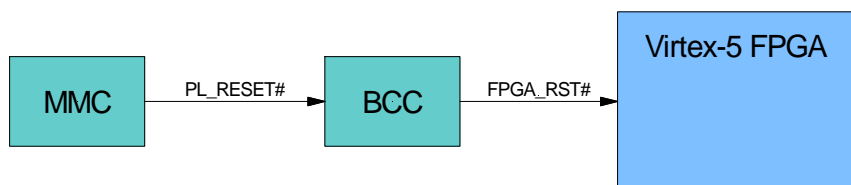


Figure 5-7 : Reset Structure

5.5 GPIO

The TAMC640 has some general purpose I/O connected to the FPGA and the CPLD.

Signal	Bank	V _{CCO}	Pin	Description
USER_SWITCH_CPLD	1	2.5V	49	Select FPGA configuration source: ON = 0x0 (SPI) OFF = 0x1 (Platform Flash)
USER_SWITCH_FPGA	21	1.8V	Y24	ON = 0x0, OFF = 0x1
GPIO_FPGA	21	1.8V	AA24	Push-Button (not installed on board, accessible via debug-connector)
PL_LED2_CTRL (CPLD)	1 (CPLD)	2.5V	AA1	Connected to MMC, pulled to MP, use as open-collector. When high, USER_LED is edge sensitive. When low, USER_LED is level sensitive
PL_LED2_1V8 (FPGA)	22	1.8V	AA2	Connected to MMC, pulled to MP, use as open-collector. Controls the AMC "USER" front panel LED2. When edge sensitive, a rising or falling edge of USER_LED triggers the MMC to turn off the USER LED in the front panel for app. 100ms. When level sensitive it directly controls the USER LED.

Table 5-4 : General Purpose I/O

5.6 I²C

The TAMC640 provides two user accessible I²C busses for communication between FPGA, FMC, Board Configuration CPLD (BCC), MMC and Si5338.

Signal	Description
SCL_PL	I ² C between FPGA, MMC, BCC and Si5338 (I ² C communication with the MMC is for future use.)
SDA_PL	
SCL_FMC_1V8	I ² C between FMC and FPGA The I ² C to the FMC allows accessing the I ² C-EEPROM on the FMC. This I ² C-bus is shared with the MMC.
SDA_FMC_1V8	
SCL_CPLD	I ² C between BCC and EEPROM Only used by the BBC for clock configuration data loading. User accessible after successful FPGA configuration.
SDA_CPLD	

Table 5-5 : I2C Bus Signals

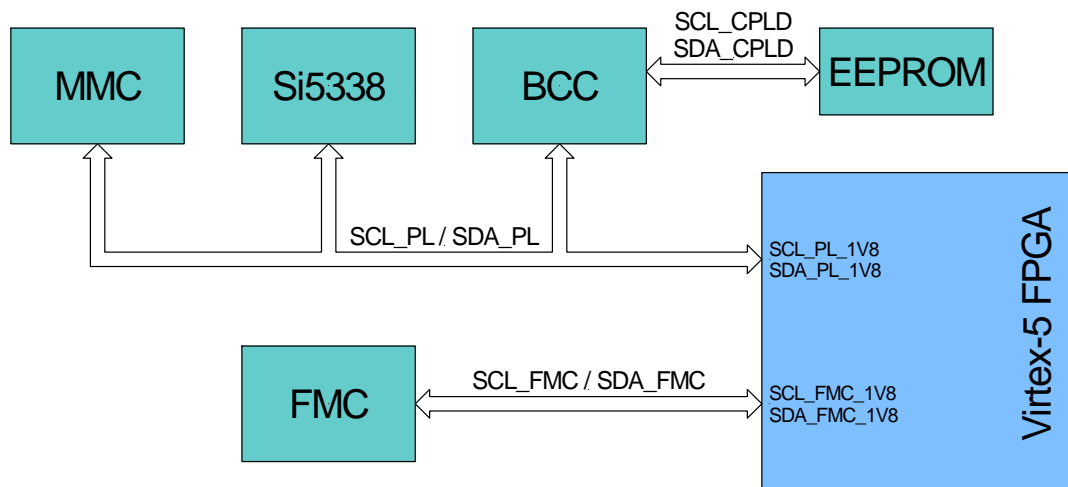


Figure 5-8 : I2C Bus Structure Overview

The MMC has a slave I²C interface, while the BCC has a master only I²C interface. The BCC dedicated I²C interface is linked to SDA_PL/SCL_PL after successful FPGA configuration. Hence, the EEPROM can also be accessed by the FPGA.

5.7 UART

Two pins of the FPGA are routed to the Debug Connector for use as debug interface (UART). This is not a real RS-232 interface. A RS-232 transceiver or USB-UART that can work with 1.8V I/O voltage should connect with these signals. TEWS TA900 provides such an interface.

Signal	V _{CC0}	Description
Rx_FPGA	1.8V	Accessible via debug-connector
Tx_FPGA	1.8V	

Table 5-6 : FPGA UART

5.8 Multi-Gigabit Transceiver (GTPs)

The TAMC640 provides 12 GTPs (also referred to as Multi Gigabit Transceivers (MGTs), or RocketIOs):

- 10 GTPs are wired to AMC ports 0, 1 (common options region) and 4 -11 (fat pipe region). FPGA hardware resources (e.g. PCI Express Endpoint Block or Gigabit Ethernet MACs) can be used with the GTPs connected to these lanes.
- 2 GTPs are wired to the FMC HPC connector (DP0 – DP1). FPGA hardware resources like the Gigabit Ethernet MACs can be used with the GTPs connected to these lanes.

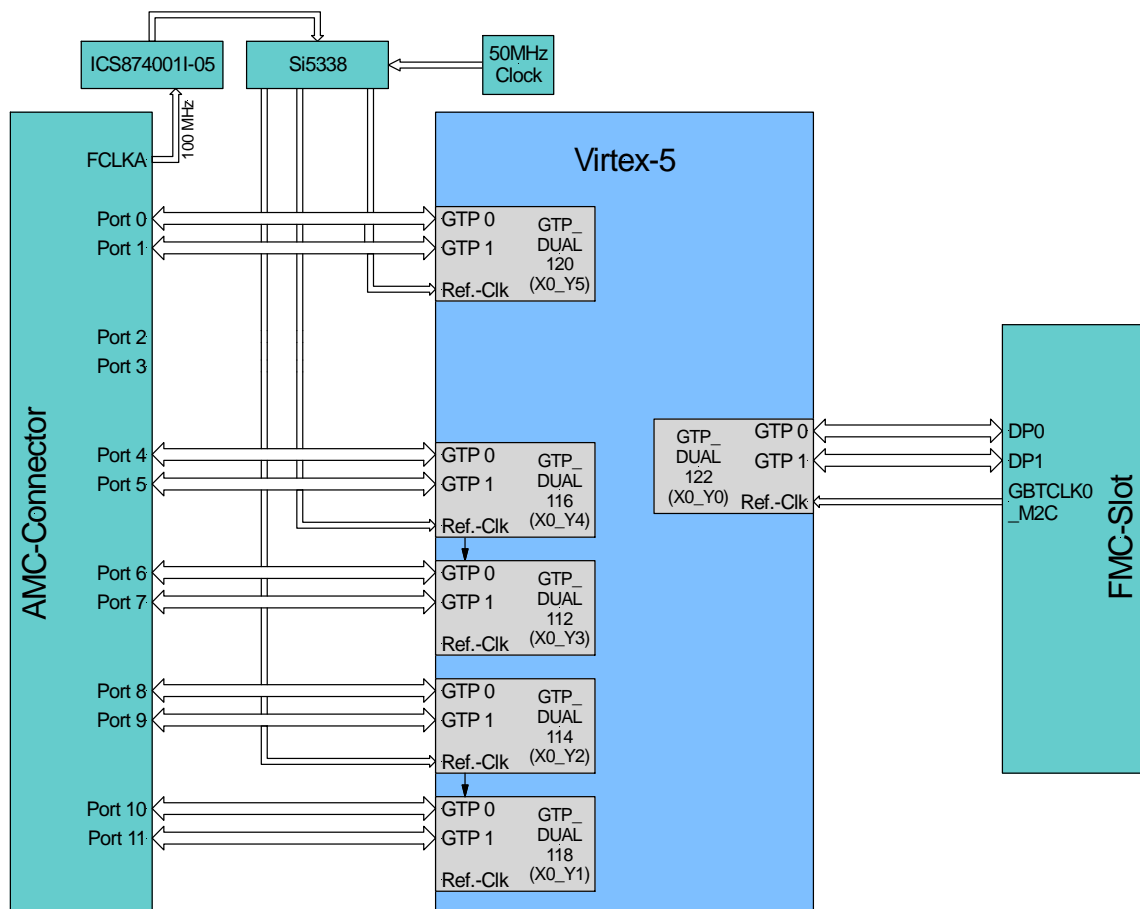


Figure 5-9 : GTP Block Diagram

5.9 Configuration

The user configurable parts of the TAMC640 are the Virtex-5 FPGA, a Board Configuration CPLD (BCC), two Xilinx Platform Flashes, a SPI-Flash and the Clock Generator device (necessary for the GTP Reference-Clock generation).

The FPGA can be configured using either of the following sources:

- Platform Flash (depending on FPGA-Size, storing multiple code revisions can be possible)
- SPI-Flash
- JTAG

The configuration method is defined by the BCC (refer next chapter). Reprogramming the BCC allows (among others) adjusting different configuration methods. Alternatively, JTAG configuration is always available. On delivery, the FPGA configuration devices are blank whereas the BCC is programmed with an initial configuration.

A green on board “DONE”-LED is lit when the FPGA is configured. If the FPGA is not configured, the red front panel out-of-service status LED remains lit.

The BCC is configured via JTAG, and handles the basic board setup.

5.9.1 Board Configuration CPLD (BCC)

The Board Configuration CPLD (BCC) is configured via JTAG, and handles the basic board setup. This setup includes

- FCLKA jitter attenuator setup
- Configuration of the GTP Reference-Clock generation (Si5338) via I²C.
- FPGA configuration source selection (performed by controlling the FPGA mode pins circuit) and starting FPGA configuration
- TCLK[A-D] M-LVDS transceiver setup

Two Platform Flashes are available. These can be used to store two or more different code versions, depending on FPGA or Code size.

An I2C EEPROM is connected to the BCC. This is used to store the GTP Reference Clock configuration data. After configuration, its I2C bus is linked to the payload I2C bus to allow the FPGA accessing the EEPROM content.

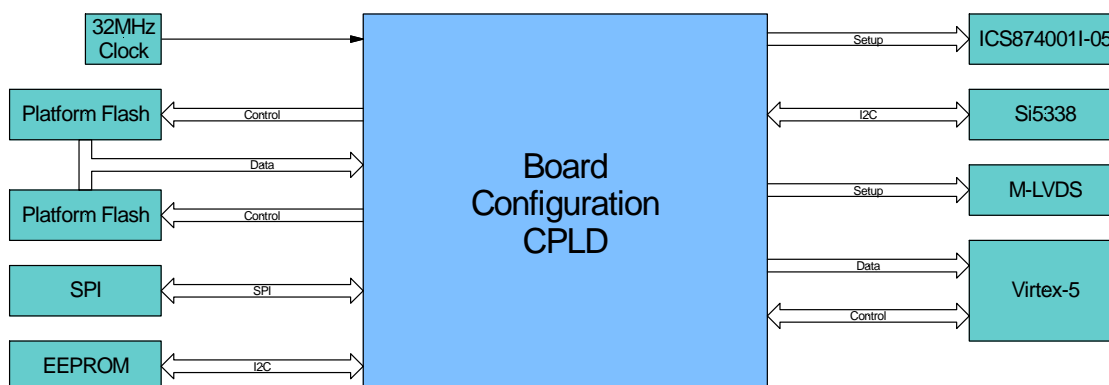


Figure 5-10: Board Configuration CPLD Block Diagram

The factory default programming of the BCC causes the following:

- MGTREFCLK_116 (GTP-Dual Tile X0Y4) driven by FCLKA (100 MHz)
- MGTREFCLK_114 (GTP-Dual Tile X0Y2) driven by locally generated 156.25 MHz clock
- MGTREFCLK_120 (GTP-Dual Tile X0Y5) driven by locally generated 125 MHz clock
- TCLK [A-D] are configured as Type 1 Inputs
- PL_LED2 is level sensitive
- FPGA Configuration Mode is “Master SelectMap” (fast configuration method) or “Master SPI”
 - o Configuration Source in “Master SelectMap” are the two cascaded Platform Flashes

The common power-up sequence is the following:

1. BCC holds PROGRAM and INIT low (delaying FPGA configuration)
2. BCC performs necessary Si5338 setup via I²C for GTP-Ref Clock generation.
3. After successful Si5338 configuration (GTP-Ref. Clocks are valid), PROGRAM followed by INIT are released to start FPGA configuration.

Configuration Source can be switched between Platform Flash and SPI-Flash by DIP-Switch 3. See chapter “Board Configuration” for more details.

After configuration, the SPI-Flash is accessible by the FPGA logic via the following pins:

FPGA Signal	FPGA Pin	Mapping
IO_L4P_FCS_B_2	AE14	SPI CS_n
CCLK_0	N15	SPI CLK
IO_L4N_VREF_FOE_B_MOSI_2	AF14	SPI MOSI
D_IN_0	P15	SPI MISO

See also Xilinx XAPP1020 “Post-Configuration Access to SPI Flash” for more details.

5.9.2 FPGA Configuration

As aforementioned, besides direct JTAG configuration, the TAMC640 provides up to three configuration sources: the two Platform Flashes and a SPI-Flash.

Configuration from the SPI-Flash is done in the Master SPI configuration mode, whereas configuration from the Platform Flash can be done in Master or Slave Serial as well as in Master or Slave SelectMap mode. There are various ways of using the Platform Flashes. All configuration devices are programmed via JTAG; the SPI-Flash uses the indirect SPI programming mode (Xilinx Impact supported method).

In all Master Modes, the Virtex-5 drives the Configuration Clock (CCLK). In all Slave Modes, the BCC drives CCLK to the Virtex-5 and the Flashes. This causes a conflict when indirect SPI programming (via Xilinx Impact) is used while the Virtex-5 is in a Slave Configuration Mode, because the Xilinx indirect SPI programming drives CCLK regardless of the FPGA Configuration Mode.



Before accessing the SPI-Flash (e.g. using Xilinx Impact), the FPGA Mode Pins must be set to “Master SPI” (factory default for V1.0 Rev. B).

If the FPGA Mode Pins are set to a Slave Configuration Mode (factory default of V1.0 Rev. A) during indirect SPI programming via Xilinx Impact, damage to the device will occur.

By default, the FPGA configures from the Platform Flash in “Master SelectMap” Mode. By the use of DIP-Switch 3, the configuration can be switched to the SPI-Flash and “Master SPI” Mode.

To change the TAMC640 programming, JTAG-capable hardware is needed (i.e. the Xilinx Platform Cable USB II).

When the SPI-Flash is used for configuration, the FPGA is always master.

Using the Platform Flashes in serial mode is not recommended, due to the high amount of time, but possible.

Performing a Platform Flash based configuration in SelectMap mode is the fastest way to get the FPGA configured. (A byte-wide interface is used in this mode.)

The following table lists the worst case configuration time of all TAMC640 configuration modes. In all Master Modes, the FPGA drives CCLK with $\pm 50\%$ frequency tolerance. The table below calculates with -50% of the nominal frequency.

Configuration			FPGA Configuration Time		
Device	Mode	max. allowed Frequency setting	LX50T	LX85T	SX50T
Platform Flash	Slave Serial	32 MHz	440 ms	730 ms	626 ms
	Master Serial	24 MHz ($\pm 50\%$)	1172 ms	1947 ms	1669 ms
	Slave SelectMap	32 MHz	55 ms	92 ms	79 ms
	Master SelectMap	20 MHz ($\pm 50\%$)	177 ms	293 ms	251 ms
SPI Flash	Master SPI	20 MHz ($\pm 50\%$)	1404 ms	2335 ms	2004 ms

Table 5-7 : Worst Case FPGA Configuration Times

In all Master configuration Modes, the desired configuration frequency must be set during bitstream generation (The Xilinx ISE “Generate Programming File” options allow to set the “Configuration Rate” (in MHz) in the “Configuration Options” category). If the Configuration Rate is not set by the user, the default configuration frequency of 2 MHz will be used and configuration time will rise up to 20 seconds.

There are different configuration options. In Master SelectMap mode, the Virtex-5 supports Fallback Multiboot. In Slave SelectMap Mode, Platform Flash decompression can be used, or fastest configuration time is achieved. The slave mode uses the on board 32 Mhz clock as configuration clock source.

Bitsream encryption is supported in all configuration modes, but not in combination with Fallback Multiboot.

After the FPGA configuration is done, the SPI-Flash is user accessible to enable the user to use it in a design, i.e. for data or code storage.

5.9.3 Clock Configuration

TCLKA – TCLKD are connected to the FPGA via M-LVDS Transceivers as single-ended nodes TCLK?_Rx and TCLK?_Tx. Transmitter enable/disable and selection of receiver input type is controlled by the Board Configuration CPLD (BCC). The Receiver path (_Rx) is always enabled.

Signal	Description
TC_DE1	Transmit-Enable for TCLKA 0 = Disable (default) 1 = Enable
TC_DE3	Transmit-Enable for TCLKB 0 = Disable (default) 1 = Enable
TC_DE0	Transmit-Enable for TCLKC 0 = Disable (default) 1 = Enable
TC_DE2	Transmit-Enable for TCLKD 0 = Disable (default) 1 = Enable
TC_FSEN2	Select receiver input type for TCLK[A&B] 0 = Type 1 receiver inputs 1 = Type 2 receiver inputs (failsafe)
TC_FSEN1	Select receiver input type for TCLK[C&D] 0 = Type 1 receiver inputs 1 = Type 2 receiver inputs (failsafe)

Table 5-8 : TCLK Transceiver configuration

FCLKA is routed through a Jitter attenuator on the TAMC640. Its configuration is also defined by the BCC. The Jitter attenuator guarantees, that the FCLKA jitter is suitable for the Virtex-5 GTP-Transceiver. The output is always enabled.

Signal	Description
J_MR	Master Reset 0 = Operation 1 = Reset
J_PLL_SEL	PLL Select 0 = Bypass PLL 1 = Use PLL (default)
J_BW_SEL	Select PLL Bandwidth 0 = 2.2 MHz (best jitter performance) 1 = 3 MHz (use for spread spectrum, factory default)
J_F_SEL[1:0]	Output-Frequency select pins 00 = 1 x Input-Frequency (factory default) 01 = 1.25 x Input-Frequency 10 = 2.5 x Input-Frequency 11 = 5 x Input-Frequency

Table 5-9 : Jitter attenuator configuration

The TAMC640 provides the Si5338 as a user programmable GTP reference-clock generator. The generator allows changing the GTP reference-clocks to any specific application needs. The Si5338 is configured at each power-up via the PL-I²C bus by the BCC. An I²C EEPROM is connected to the BCC as non-volatile clock-configuration-data storage. At power up, the BCC reads the configuration data from the EEPROM, and writes it to the Si5338. The Si5338 INTR interrupt line can be used to detect e.g. a Loss of clock.

By default, this signal is not used in the factory default CPLD code.

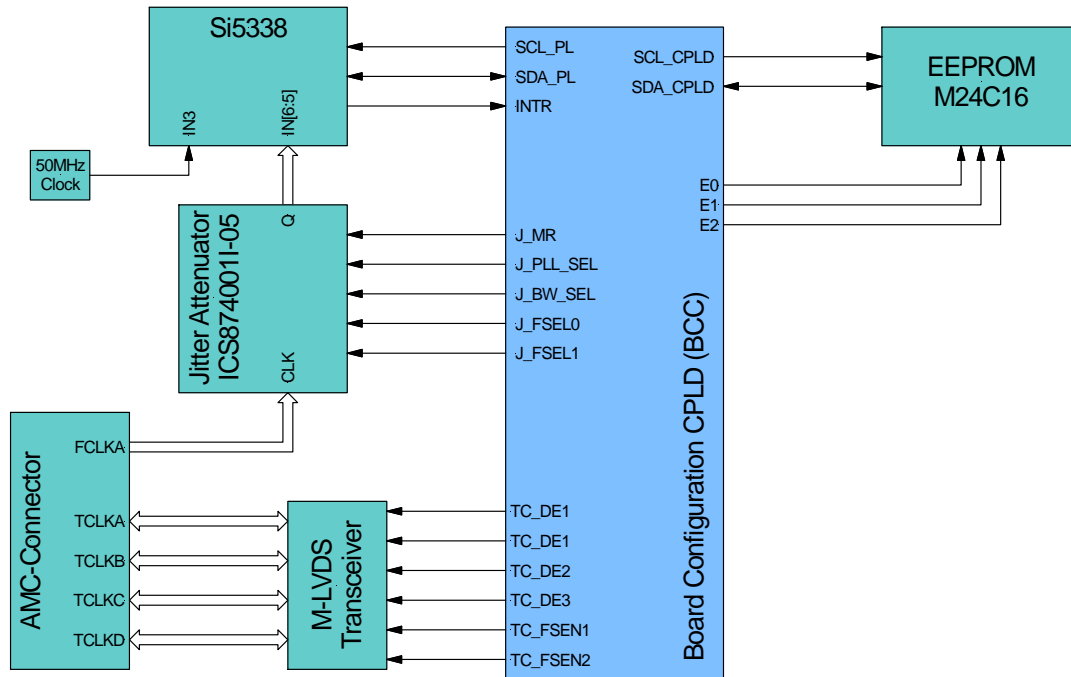


Figure 5-11: Clock Configuration

5.10 Clocks

The TAMC640 has the following main clock sources:

- 100 MHz AMC fabric clock FCLKA. Routed through an ICS874001I-05 PCI Express jitter-attenuator, which feeds the Si5338 that generates up to four clocks of any frequency needed. These clocks are connected to GTP reference clock inputs.
- 50 MHz, provided by two fixed frequency oscillators. These clocks are connected to global clock inputs of the FPGA.
- Clocks provided by the FMC slot. These clocks are connected to global and regional (clock capable) clock pins of the Virtex-5 FPGA, except GBTCLK0_M2C and GBTCLK1_M2C, which connect to GTP reference clock inputs.
- TCLK[A-D]. All four AMC TCLK signals are connected via single-ended nodes to global clock pins of the FPGA via M-LVDS transceivers. The M-LVDS transceiver can independently be configured as input or output.

AMC FCLKA is connected to the FPGA via a PCIe jitter-attenuator and a clock generator (Si5338) to reduce the Clock Jitter and allow pre-scaling the clock. The PCI-Express Interface works with Spread-Spectrum Clock (SSC) and non SSC PCI-Express Reference Clocks.

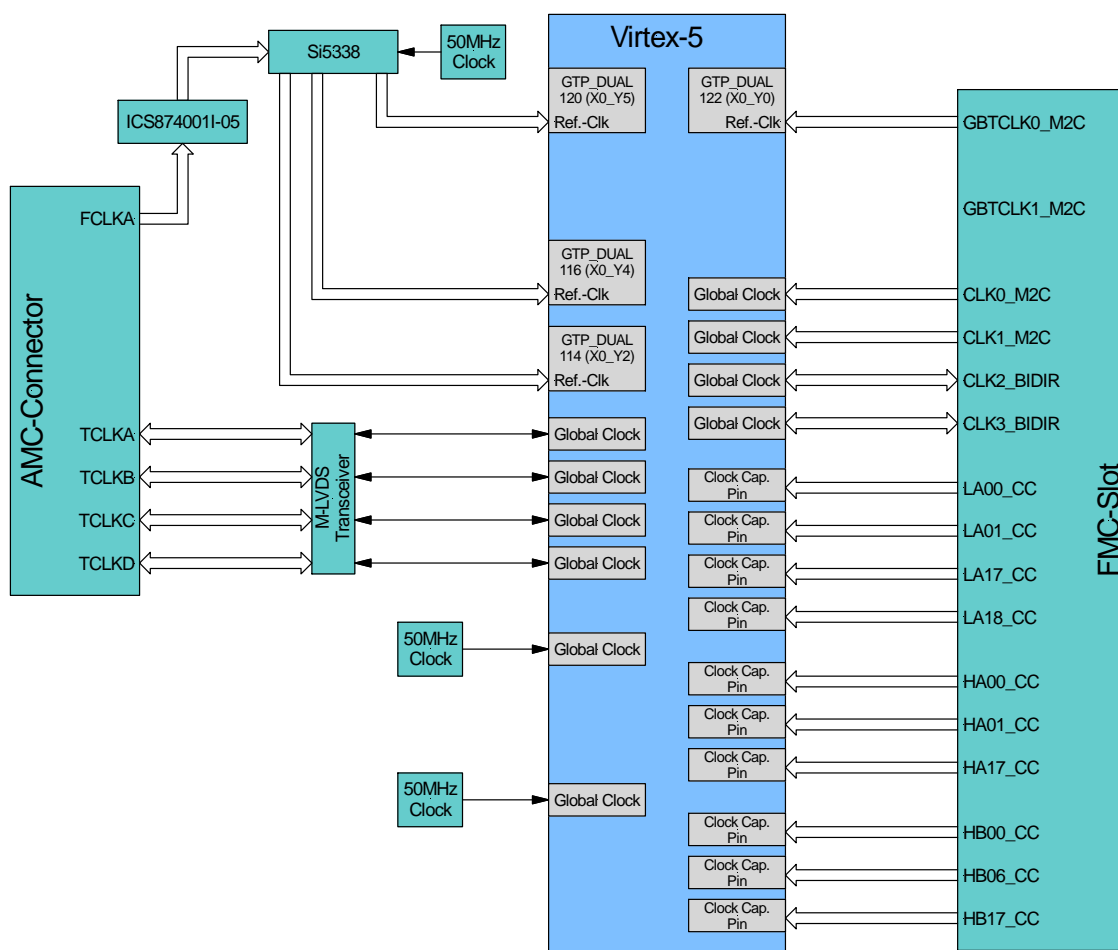


Figure 5-12: FPGA Clock Sources

The following table lists the available clock sources on the TAMC640:

FPGA Clock Pin/Signal Name	FPGA Pin Number	Source	Description
MGTREFCLK_120	E4 / D4	Si5338 CLK0	GTP Ref.-Clock, differential
MGTREFCLK_116	H4 / H3	Si5338 CLK2	GTP Ref.-Clock, differential
MGTREFCLK_114	Y4 / Y3	Si5338 CLK3	GTP Ref.-Clock, differential
MGTREFCLK_122	AL5 / AL4	FMC GBTCLK0_M2C	GTP Ref.-Clock (from FMC) , differential
TCLKA_Rx	AH20	TCLKA	Via M-LVDS Transceiver
TCLKB_Rx	AH14	TCLKB	Via M-LVDS Transceiver
TCLKC_Rx	AG22	TCLKC	Via M-LVDS Transceiver
TCLKD_Rx	AH12	TCLKD	Via M-LVDS Transceiver
UCLK	AG21	On board 50MHz Oscillator	
HA_CLK	H17	On board 50MHz Oscillator	AC-Coupled
CLK0_M2C	AF18 / AE18	FMC CLK0_M2C	LPC & HPC FMC, differential
CLK1_M2C	AH18 / AG17	FMC CLK1_M2C	LPC & HPC FMC, differential
CLK2_BIDIR	AG18 / AF19	FMC CLK2_BIDIR	HPC FMC only, differential, driven by either FPGA or FMC
CLK3_BIDIR	AH17 / AG16	FMC CLK3_BIDIR	HPC FMC only, differential, driven by either FPGA or FMC
LA00_CC	AH34 / AJ34	FMC LA00_CC	LPC & HPC FMC
LA01_CC	AF34 / AE34	FMC LA01_CC	LPC & HPC FMC
LA17_CC	K33 / K32	FMC LA17_CC	LPC & HPC FMC
LA18_CC	L34 / K34	FMC LA18_CC	LPC & HPC FMC
HA00_CC	M31 / N30	FMC HA00_CC	HPC FMC only
HA01_CC	P31 / P30	FMC HA01_CC	HPC FMC only
HA17_CC	K17 / L18	FMC HA17_CC	HPC FMC only
HB00_CC	G27 / H27	FMC HB00_CC	HPC FMC only
HB06_CC	H28 / G28	FMC HB06_CC	HPC FMC only
HB17_CC	G23 / H23	FMC HB17_CC	HPC FMC only

Table 5-10: Available FPGA clocks

5.10.1 GTP Reference Clock Generator

The TAMC640 provides a user programmable Si5338 clock generator. The clock generator allows changing the GTP Reference-Clocks to any specific application needs.

AMC FCLKA (lead over the on board Jitter Attenuator) or the on board 50 MHz clock can be used as clock source for the GTP reference clock generation.

Silicon Labs supplies software, which can be used to generate the settings for the desired GTP reference clocks. The default clock settings are:

Pin	Frequency	FPGA Pin	Description
IN5 / IN6	100 MHz	-	Clock generator differential input, connected to Jitter Attenuator / PCIe Reference Clock
IN3	50MHz	-	On board clock
CLK0	125 MHz (default)	Ref-Clkin of GTP_DUAL_X0Y5	Reference Clock for AMC Port 0 & 1
CLK1	150 MHz (default)	-	not used
CLK2	100 MHz (default)	Ref-Clkin of GTP_DUAL_X0Y4	Reference Clock for AMC Port 4 – 7
CLK3	156.25 MHz (default)	Ref-Clkin of GTP_DUAL_X0Y2	Reference Clock for AMC Port 8 – 11

Table 5-11: Programmable GTP Reference Clock

For an instruction on how to reprogram the clock generator, refer to chapter “GTP Reference Clock Generator Configuration”.

5.11 JTAG

Beneath Platform Flash and SPI-Flash programming, direct FPGA configuration, FPGA readback or in-system diagnostics with ChipScope is possible using the JTAG-chain. The JTAG-chain can be extended to include the FMC-Slot, so JTAG capable FMCs can be accessed.

The JTAG-chain is either accessible from the Debug Connector or from the AMC backplane JTAG port. If a debug adapter is connected to the TAMC640, the AMC backplane JTAG port is disabled. To ease the use of the JTAG-chain, it is partitioned into segments. Each segment can be separately held inactive and thereby excluded (“bypassed”) from the chain. This allows masking the on board JTAG devices segment when a JTAG-device on a mounted FMC is targeted and vice versa.

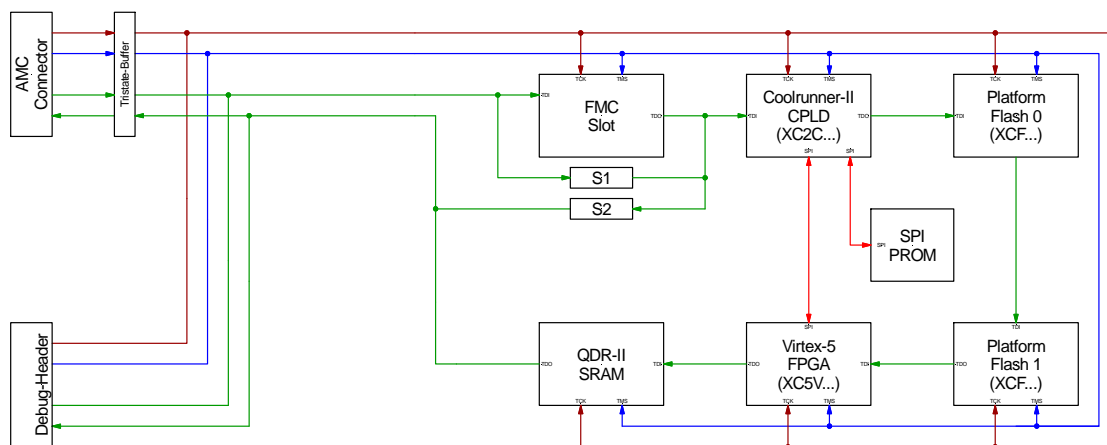


Figure 5-13: JTAG-Chain Segmentation

The FMC-Slot is only included into the JTAG chain when a FMC is installed (PRSNT_M2C# is asserted). (Signal not shown in the preceding figure.)

The Configuration DIP-Switch allows configuring the JTAG-chain. The Configuration DIP-Switch is located on the back side of the TAMC640.

Switch	Signal	Description
SW1	ON	Include FMC in JTAG-chain
	OFF	Bypass FMC
SW2	ON	Include TAMC640 devices in JTAG-chain
	OFF	Bypass TAMC640 devices

Table 5-12: Configuration DIP-Switch SW1-SW2 Settings

Devices in inactive segments are held in the Test-Logic-Reset State.

5.12 Thermal Management

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

The TAMC640 has a heatsink mounted on the Virtex-5 FPGA. The heatsink provides a R_{TH} of app. 6 K/W without air flow, with forced air cooling R_{TH} will decrease to app. 1.5 K/W.

6 Board Configuration

This chapter describes aspects of board configuration prior to board installation.

6.1 Overview

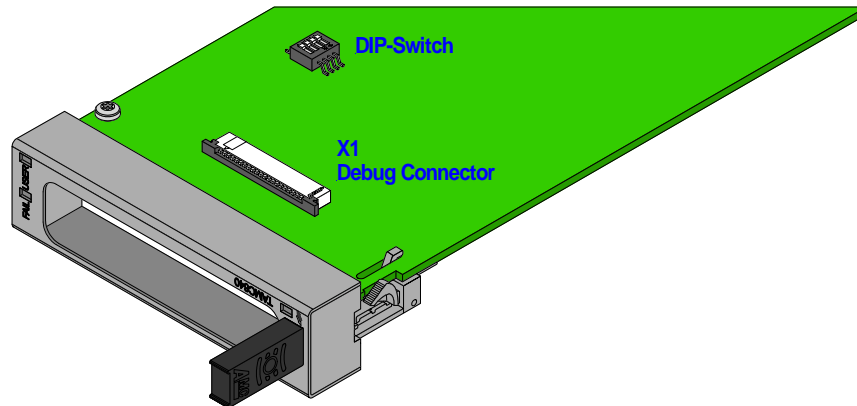


Figure 6-1 : Pre-Insertion Board Configuration Overview

6.2 DIP-Switch

The DIP-Switch is located on the bottom side of the TAMC640 and provides the following configuration options.

Switch No.	Switch Position	Description
1	ON	Include FMC in the JTAG-chain (default)
	OFF	Bypass FMC JTAG devices
2	ON	Include TAMC640 devices in the JTAG-chain (default)
	OFF	Bypass TAMC640 JTAG devices
3	ON	FPGA configures from SPI-Flash
	OFF	FPGA configures from Platform Flash (default)
4	ON	USER_SWITCH_FPGA read as 0
	OFF	USER_SWITCH_FPGA read as 1 (default)

Table 6-1 : DIP-Switch

6.3 Battery

Virtex-5 devices have on-chip decryption logic to support encrypted FPGA bitstream usage. Encrypted FPGA bitstreams cannot be copied or reverse engineered, securing your intellectual property.

The TAMC640 provides a retainer for a 1225 button/coin cell. This battery is only used to store the encryption key inside the FPGA. To enable the usage of FPGA bitstream encryption, a battery (1225 button/coin cell) has to be populated in the TAMC640 battery retainer:

1. Remove the heat sink
2. Insert battery into holder
3. Remount heat sink

The thermal interface material between FPGA and Heat Sink is a so called “phase change material”, and does not need to be renewed after heat sink removal.

6.4 Debug Connector

The Debug Connector (X1) of the TAMC640 can be used to connect a Debug Adapter if necessary. The debug Adapter must be connected to the TAMC640 prior to AMC installation. It is recommended to use the TEWS TA900 Debug Adapter.

The Debug Connector provides four logical interfaces: JTAG, MMC-UART, FPGA-UART and a General Purpose User Signal (GPIO_BUT).

- The JTAG interface consists of the signals TDI, TDO, TMS, TCK, uses 3.3V I/O voltage, and can run with up to 10 MHz.
- The MMC-UART consists of Rx and Tx, uses 3.3V I/O voltage, and runs at 38400 bit/s using 8 data bits, no parity, one stop bit.
- The FPGA-UART consists of Rx and Tx and uses 1.8V I/O voltage. Communication settings depend on the FPGA programming.
- The General Purpose User Signal uses 1.8V I/O voltage. When used with the TEWS TA900, this signal is connected to a Push button on the TEWS TA900 and must be configured as FPGA input.

7 Installation

This chapter contains general notes regarding installing the AMC module into a system.

7.1 Installation of a FMC Module

Before installing a FMC module, make sure that the power supply for the TAMC640 is turned off.

Components are Electrostatic Sensitive Devices (ESD). Use an anti-static mat connected to a wristband when handling or installing the components.

7.1.1 Using FMCs with Mid-Size faceplates

The TAMC640 places the FMC directly at the AMC faceplate. The TAMC640 Mid-Size faceplate provides a cut-out to ease the installation of the FMC to the TAMC640. Pins of FMC I/O-connectors that protrude on the Side 2 (the “back side”) of the FMC (compare read arrows) may still touch the AMC front panel. This is a potential hazardous electrical problem, depending on the I/O circuitry used.

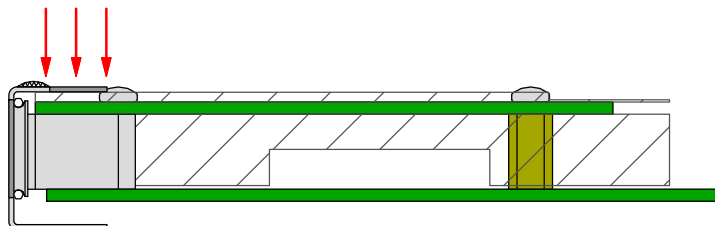


Figure 7-1 : Using FMCs with Mid-Size faceplates

It is within the responsibility of the user to carefully check whether a specific FMC can be used on a Mid-Size TAMC640. When you are not sure that the available spacing to conductive parts of the FMC is sufficient, it is strongly recommended to use a TAMC640 with Full-Size front panel.

7.1.2 Voltage Limits on FMC Modules

The AMC.0 specification limits the voltages on AMC modules to following thresholds:

	DC voltage	AC voltage
Positive	+27V	+27V peak
Negative	-15V	-15V peak

Table 7-1 : Voltage Limits on FMC Modules

For FMC modules using voltages (including I/O voltages) that exceed these thresholds, an additional insulation to adjacent modules or carrier boards becomes necessary.

7.2 AMC Module Insertion & Hot-Swap

During insertion and extraction, the operational state of the AMC is visible via the blue LED in the AMCs front panel. The following table lists all valid combinations of Hot-swap handle position and blue LED status, including a short description of what's going on.

Blue LED \ Handle	On	Off	Long Blink	Short Blink
Open (Pulled out)	Extraction: Module can be extracted Insertion: Module is waiting for closed Handle	Module is waiting for hot swap negotiation	-	Hot swap negotiation in progress (Extraction)
Closed (Pushed all way in)	Module is waiting for hot swap negotiation	Module is active (operating)	Hot swap negotiation in progress (Insertion)	-

Table 7-2 : Hot-Swap states

7.2.1 Insertion

Typical insertion sequence:

1. Insert the AMC module into an appropriate slot, with the board edges aligned to the card guides
2. Make sure that the module handle is pushed positional way in
 - a. Blue LED turns "ON." (Module is ready to attempt activation by the system)
 - b. Blue LED starts "Long Blink" (Hot Swap Negotiation / Module activation in progress)
 - c. Blue LED turns "OFF", and green LED turns "ON" (Module is ready and powered)

When the Blue LED does not go off but returns to the "ON" state, the module FRU information is invalid or the system cannot provide the power requested by the AMC module.

If the blue LED is off, but the red front panel out-of-service status LED remains lit, the FPGA may not be configured.

7.2.2 Extraction

Typical extraction sequence:

1. Pull the module handle out half way out
 - a. Blue LED starts "Short Blink" (Hot Swap Negotiation in progress)
 - b. Blue LED turns "ON" (Module is ready to be extracted)
2. Pull the module handle out completely and extract the AMC module from the slot.

8 Indicators

This chapter describes all board indicators (LEDs) of the TAMC640.

8.1 Front Panel LEDs

For a quick visual status inspection, the AMC module provides the following front panel LEDs.

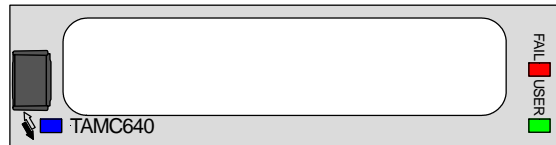


Figure 8-1 : Front Panel LED View

LED	Color	State	Description
HS	Blue	Off	No Power or Module is ready for normal operation
		Short Blink	Hot-Swap negotiation (extraction)
		Long Blink	Hot-Swap negotiation (insertion)
		On	Module is ready to attempt activation by the system or Module is ready to be extracted
FAIL	Red	Off	No fault
		On	Failure or out of service status
USER	Green	Off	Design dependent, can be controlled by the FPGA. Refer to chapter “GPIO”
		On	
		Blink	

Table 8-1 : Front Panel LEDs

8.2 On board LEDs

The TAMC640 provides a couple of board-status LEDs as shown below. These include Power-Good and FPGA configuration status indications.

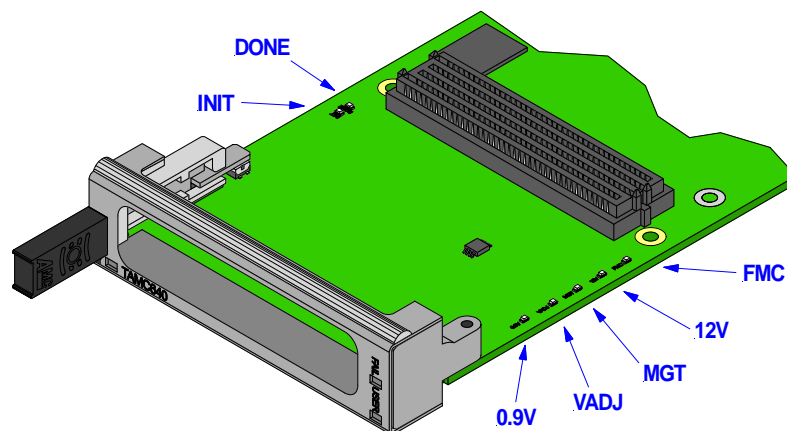


Figure 8-2 : On board LED View

Indicator	Color	Description
FMC	Green	Power Good from FMC to Carrier Card Indicates that all FMC supplies are within tolerance
12V	Green	Power Good for +12V FMC supply
MGT	Green	Power Good for FPGA supplies
VADJ	Green	Power Good for VADJ
0V9	Green	Power Good for QDR-II and DDR2 power supplies.
DONE	Green	FPGA DONE-Pin LED Indicates successful FPGA configuration
INIT#	Red	FPGA INIT-Pin LED DONE Low : Indicates unsuccessful FPGA configuration DONE High : Readback CRC Error if Readback CRC is enabled

Table 8-2 : Board-Status LEDs

9 I/O Connectors

9.1 Overview

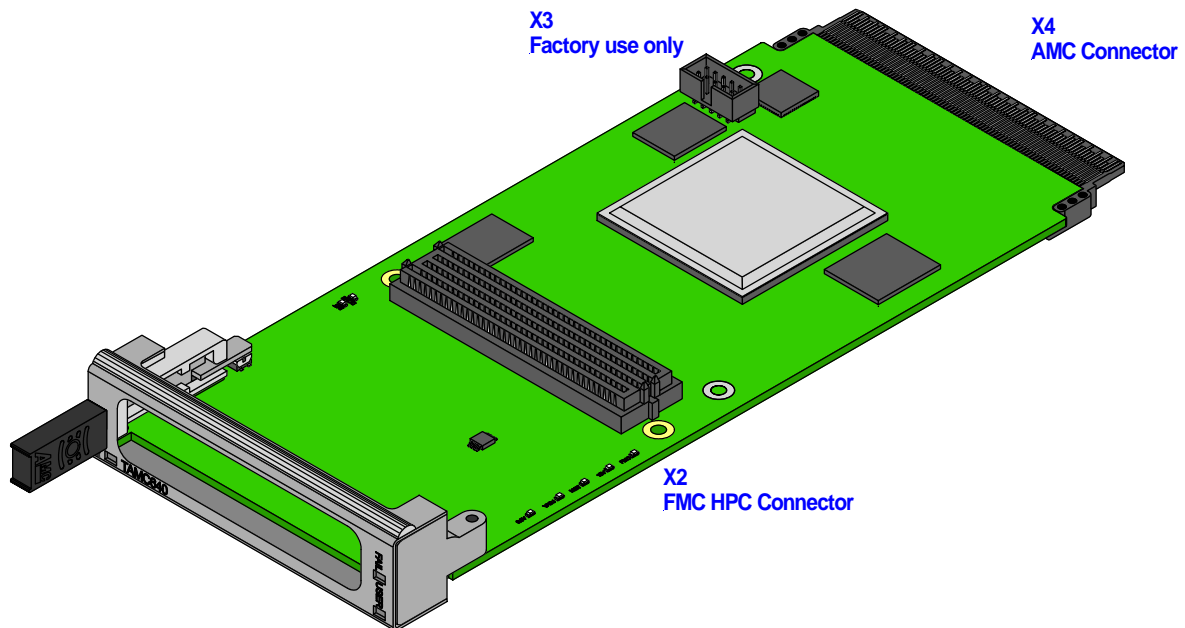


Figure 9-1: Connector Positions – Side 1

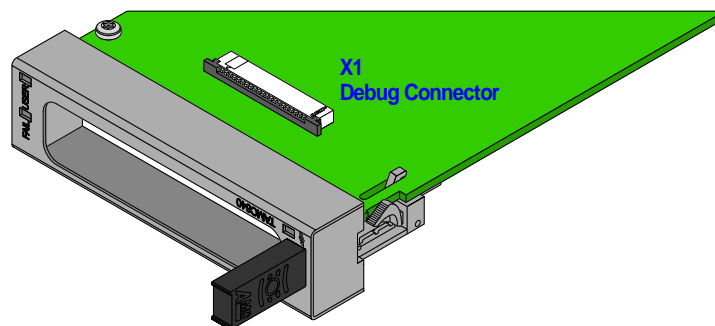


Figure 9-2: Connector Position – Side 2

9.2 I/O Circuitry

All FMC I/O lines are directly connected to the FPGA-pins. Together with the adjustable V_{CC0} and V_{REF} this maintains the flexibility of the SelectIO technology of the Virtex-5 FPGA. Refer to UG190: *Virtex-5 FPGA User Guide* for SelectIO interface signal standards, slew rate control and current drive strength capabilities.

9.2.1 Differential Signaling

As defined in the FMC specification, the TAMC640 expects the AC-coupling for DP signals to be placed on the FMC.

9.3 AMC-Connector X4

This is an excerpt of the AMC-connector pin assignment. Only the user available signals are listed.

Pin	Signal	Function
15	Rx0-	AMC port 0 (normally used for GbE)
14	Rx0+	
12	Tx0-	Connected to GTP_DUAL_X0Y4
11	Tx0+	
24	Rx1-	AMC port 1 (normally used for GbE)
23	Rx1+	
21	Tx1-	Connected to GTP_DUAL_X0Y4
20	Tx1+	
33	Rx2-	-
32	Rx2+	
30	Tx2-	
29	Tx2+	
39	Rx3-	-
38	Rx3+	
36	Tx3-	
35	Tx3+	
48	Rx4-	AMC port 4 (used for PCIe, SRIO, XAUI,...)
47	Rx4+	
45	Tx4-	Connected to GTP_DUAL_X0Y3
44	Tx4+	
54	Rx5-	AMC port 5 (used for PCIe, SRIO, XAUI,...)
53	Rx5+	
51	Tx5-	Connected to GTP_DUAL_X0Y3
50	Tx5+	
63	Rx6-	AMC port 6 (used for PCIe, SRIO, XAUI,...)
62	Rx6+	
60	Tx6-	Connected to GTP_DUAL_X0Y2
59	Tx6+	
69	Rx7-	AMC port 7 (used for PCIe, SRIO, XAUI,...)
68	Rx7+	
66	Tx7-	Connected to GTP_DUAL_X0Y2
65	Tx7+	
81	FCLKA-	Fabric Clock (100MHz)
80	FCLKA+	

Pin	Signal	Function
87	Rx8-	AMC port 8 (used for PCIe, SRIO, XAUI,...)
88	Rx8+	
90	Tx8-	Connected to GTP_DUAL_X0Y1
91	Tx8+	
93	Rx9-	AMC port 9 (used for PCIe, SRIO, XAUI,...)
94	Rx9+	
96	Tx9-	Connected to GTP_DUAL_X0Y1
97	Tx9+	
99	Rx10-	AMC port 10 (used for PCIe, SRIO, XAUI,...)
100	Rx10+	
102	Tx10-	Connected to GTP_DUAL_X0Y0
103	Tx10+	
105	Rx11-	AMC port 11 (used for PCIe, SRIO, XAUI,...)
106	Rx11+	
108	Tx11-	Connected to GTP_DUAL_X0Y0
109	Tx11+	
111	Rx12-	AMC Port 12
112	Rx12+	
114	Tx12-	
115	Tx12+	
117	Rx13-	AMC Port 13
118	Rx13+	
120	Tx13-	
121	Tx13+	
123	Rx14-	AMC Port 14
124	Rx14+	
126	Tx14-	
127	Tx14+	
129	Rx15-	AMC Port 15
130	Rx15+	
132	Tx15-	
133	Tx15+	
141	Rx17-	AMC Port 17
142	Rx17+	
144	Tx17-	
145	Tx17+	

Pin	Signal	Function	Pin	Signal	Function
78	TCLKB-	Differential Clock	135	TCLKC-	Differential Clock
77	TCLKB+		136	TCLKC+	
75	TCLKA-	Differential Clock	138	TCLKD-	Differential Clock
74	TCLKA+		139	TCLKD+	

Table 9-1: Pin Assignment AMC Connector X4

9.4 FMC HPC Connector X2

The TAMC640 provides a High Pin Count interface. The connector is a Samtec #ASP-134486-01.

Pin	K	J	H	G	F
1	VREF_B_M2C	GND	VREF_B_M2C	GND	PG_M2C
2	GND	CLK3_M2C_P	PRSNT_M2C_L	CLK1_M2C_P	GND
3	GND	CLK3_M2C_N	GND	CLK1_M2C_N	GND
4	CLK2_M2C_P	GND	CLK0_M2C_P	GND	HA00_P_CC
5	CLK2_M2C_N	GND	CLK0_M2C_N	GND	HA00_N_CC
6	GND	HA03_P	GND	LA00_P_CC	GND
7	HA02_P	HA03_N	LA02_P	LA00_N_CC	HA04_P
8	HA02_N	GND	LA02_N	GND	HA04_N
9	GND	HA07_P	GND	LA03_P	GND
10	HA06_P	HA07_N	LA04_P	LA03_N	HA08_P
11	HA06_N	GND	LA04_N	GND	HA08_N
12	GND	HA11_P	GND	LA08_P	GND
13	HA10_P	HA11_N	LA07_P	LA08_N	HA12_P
14	HA10_N	GND	LA07_N	GND	HA12_N
15	GND	HA14_P	GND	LA12_P	GND
16	HA17_P_CC	HA14_N	LA11_P	LA12_N	HA15_P
17	HA17_N_CC	GND	LA11_N	GND	HA15_N
18	GND	HA18_P	GND	LA16_P	GND
19	HA21_P	HA18_N	LA15_P	LA16_N	HA19_P
20	HA21_N	GND	LA15_N	GND	HA19_N
21	GND	HA22_P	GND	LA20_P	GND
22	HA23_P	HA22_N	LA19_P	LA20_N	HB02_P
23	HA23_N	GND	LA19_N	GND	HB02_N
24	GND	HB01_P	GND	LA22_P	GND
25	HB00_P_CC	HB01_N	LA21_P	LA22_N	HB04_P
26	HB00_N_CC	GND	LA21_N	GND	HB04_N
27	GND	HB07_P	GND	LA25_P	GND
28	HB06_P_CC	HB07_N	LA24_P	LA25_N	HB08_P
29	HB06_N_CC	GND	LA24_N	GND	HB08_N
30	GND	HB11_P	GND	LA29_P	GND
31	HB10_P	HB11_N	LA28_P	LA29_N	HB12_P
32	HB10_N	GND	LA28_N	GND	HB12_N
33	GND	HB15_P	GND	LA31_P	GND
34	HB14_P	HB15_N	LA30_P	LA31_N	HB16_P
35	HB14_N	GND	LA30_N	GND	HB16_N
36	GND	HB18_P	GND	LA33_P	GND

Pin	K	J	H	G	F
37	HB17_P_CC	HB18_N	LA32_P	LA33_N	HB20_P
38	HB17_N_CC	GND	LA32_N	GND	HB20_N
39	GND	VIO_B_M2C	GND	VADJ	GND
40	VIO_B_M2C	GND	VADJ	GND	VADJ

Table 9-2: Pin Assignment FMC-Connector X2 Row F-K

Pin	E	D	C	B	A
1	GND	PG_M2C	GND	CLK_DIR	GND
2	HA01_P_CC	GND	DP0_C2M_P	GND	DP1_M2C_P
3	HA01_N_CC	GND	DP0_C2M_N	GND	DP1_M2C_N
4	GND	GBTCLK0_M2C_P	GND	-	GND
5	GND	GBTCLK0_M2C_N	GND	-	GND
6	HA05_P	GND	DP0_M2C_P	GND	-
7	HA05_N	GND	DP0_M2C_N	GND	-
8	GND	LA01_P_CC	GND	-	GND
9	HA09_P	LA01_N_CC	GND	-	GND
10	HA09_N	GND	LA06_P	GND	-
11	GND	LA05_P	LA06_N	GND	-
12	HA13_P	LA05_N	GND	-	GND
13	HA13_N	GND	GND	-	GND
14	GND	LA09_P	LA10_P	GND	-
15	HA16_P	LA09_N	LA10_N	GND	-
16	HA16_N	GND	GND	-	GND
17	GND	LA13_P	GND	-	GND
18	HA20_P	LA13_N	LA14_P	GND	-
19	HA20_N	GND	LA14_N	GND	-
20	GND	LA17_P_CC	GND	-	GND
21	HB03_P	LA17_N_CC	GND	-	GND
22	HB03_N	GND	LA18_P_CC	GND	DP1_C2M_P
23	GND	LA23_P	LA18_N_CC	GND	DP1_C2M_N
24	HB05_P	LA23_N	GND	-	GND
25	HB05_N	GND	GND	-	GND
26	GND	LA26_P	LA27_P	GND	-
27	HB09_P	LA26_N	LA27_N	GND	-
28	HB09_N	GND	GND	-	GND
29	GND	TCK	GND	-	GND
30	HB13_P	TDI	SCL	GND	-
31	HB13_N	TDO	SDA	GND	-
32	GND	3P3V AUX	GND	-	GND

Pin	E	D	C	B	A
33	HB19_P	TMS	GND	-	GND
34	HB19_N	TRST_L	GA0	GND	-
35	GND	GA1	12V	GND	-
36	HB21_P	3P3V	GND	-	GND
37	HB21_N	GND	12V	-	GND
38	GND	3P3V	GND	GND	-
39	VADJ	GND	3P3V	GND	-
40	GND	3P3V	GND	-	GND

Table 9-3: Pin Assignment FMC-Connector X2 Row A-E

9.5 Debug-Connector X1

Pin	Signal	I/O	Description
1	JTAG_SEL	O	A 1k pullup to 3.3 Volt is located on the TAMC640
2	3.3V	O	JTAG reference I/O voltage
3	TDO	O	Test Data Output
4	GND	-	Ground
5	TDI	I	Test Data Input
6	TMS	I	Test Mode Select Input
7	GND	-	Ground
8	TCK	I	Test Clock
9	GND	-	Ground
10	UART_RxD	I	FPGA UART Receive Data
11	1.8V	O	UART reference I/O voltage
12	UART_TxD	O	FPGA UART Transmit Data (driven by FPGA)
13	GND	-	Ground
14	MMC_RxD	I	MMC UART Receive Data
15	MP	O	UART reference I/O voltage (3.3V)
16	MMC_TxD	O	MMC UART Transmit Data (driven by MMC)
17	GND	-	Ground
18	3.3V	O	+3.3 Volt
19	1.8V	O	User signal reference I/O voltage
20	GPIO_BUT	I	User signal connected to the FPGA. Use FPGA internal Pullup if needed.

Table 9-4: Pin Assignment Debug Connector X1

10 Design Help

10.1 GTP Reference Clock Generator Configuration

Use the Silicon Labs software to generate a new configuration file.

Take the related register content, and program it into the EEPROM.

10.2 Example Design

TEWS offers an FPGA Development Kit (TAMC640-FDK), which consists of a well documented basic example design. It includes an .ucf file with all necessary pin assignments and basic timing constraints. The example design covers the main functionalities of the TAMC640. It implements a DMA capable PCIe endpoint with interrupt support, register mapping, DDR2 and QDR-II memory access and basic I/O to the FMC slot. It comes as a Xilinx ISE project with source code and as a ready-to-download bitstream. This example design can be used as a starting point for own projects.

The basic example design requires the Embedded Development Kit (EDK), which is part of the Embedded or System Edition of the ISE Design Suite from Xilinx (downloadable from www.xilinx.com, a 30 day evaluation license is available) or can be licensed separately. **It will not work with the free ISE WebPACK.**

10.3 Troubleshooting

10.3.1 Board does not power up

Possible Cause:

- (1) An FMC without valid EEPROM content is mounted, and the MMC is not able to set V_{ADJ} . To avoid potential hardware damage, the board does not power up.
A solution is to manually set the VADJ as described in chapter “Internal Use Area”.
- (2) The module current requirements including the FMC exceed the system limits.
There are several possibilities to solve this issue:
 - a. Remove other AMCs from the system
 - b. Use a different power-supply with higher wattage
 - c. Modify the “Current Draw” value in the Multi record Area, as described in chapter “Module Current Requirements”

10.3.2 DONE is always off

Possible Cause:

- (1) If the Configuration Clock Rate for the FPGA bitstream generation is left at the Xilinx default setting of 2 MHz and the SPI flash is selected as configuration source, it may take up to one minute until the FPGA configuration is finished.
Set the Configuration Clock Rate to the desired value as described in chapter “FPGA Configuration” to speed up your FPGA configuration. Do not exceed the maximum value for the Configuration Clock Rate. Otherwise the FPGA configuration will fail.
- (2) The configuration source is empty or the wrong configuration source is selected.
Make sure that you programmed your bitstream into the desired configuration source, select the correct configuration source via DIP Switch 3 and try again.
- (3) The board is shipped with a preprogrammed Board Configuration CPLD (BCC), which is necessary for FPGA configuration. If the BCC is erased by mistake, FPGA configuration will fail. You have to reprogram the BCC for successful FPGA configuration and board operation. The factory default BCC program-file is part of the TAMC640-ED and the TAMC640-FDK.

10.3.3 INIT LED stays illuminated (red)

Possible Cause:

- (1) If the Si5338 configuration data is modified by the customer, the Si5338 configuration may fail due to faulty settings.
Carefully check your custom Si5338 settings, or do a cross-check with the factory default settings.
- (2) FPGA indicates a CRC or IDCODE Error during configuration (DONE LED off).
Please check if you selected the correct Device and Package in your VHDL-project design flow.

11 Appendix A

This appendix contains the signal to pin assignments for the Board Configuration CPLD (BCC).

```
## ##### ##
##                                     TEWS TECHNOLOGIES                                     ##
## ##### ##
##
## Project Name      : TAMC64x Configuration CPLD
## File Name        : tamc64x_cpld.ucf
## Target Device    : XC2C256-6VQ100
## Design Tool      : Xilinx ISE Design Suit Embedded 13.2
## Simulation Tool  : Xilinx ISIM
##
## Description      : The files lists all CPLD pins that are connected on the TAMC64x
##
## Owner           : TEWS TECHNOLOGIES GmbH
##                  Am Bahnhof 7
##                  D-25469 Halstenbek
##
##                  Tel.: +49 / (0)4101 / 4058-0
##                  Fax.: +49 / (0)4101 / 4058-19
##                  e-mail: support@tews.com
##
##                  Copyright (c) 2011
##                  TEWS TECHNOLOGIES GmbH
##
## History          :
##   Version 1     : (SE, 13.04.2011)
##                   Initial Version
##   Version 2     : (SE, 04.07.2011)
##                   For safety reason the following changes have been made:
##                   - Set timing constraints for PAD-to-PAD I/O
##                   - Set timing constraints for CPLD payload and EEPROM I2C bus
##                   - Added pull up constraints for SPI_MISO, FPGA_RS[x], CFG_D[5,6], FCS_n, CCLK,
##                     MOSI and D_IN
##   Version 3     : (SE, 05.07.2011)
##                   For safety reason the following changes have been made:
##                   - Added Schmitt Trigger I/O for FCS_n and DONE
##                   - Removed MMC SPI and Revision Interface
##   Version 4     : (SE, 08.07.2011)
##                   Changed Design Tool
##   Version 5     : (SE, 11.07.2011)
##                   Revised timing constraints
##   Version 6     : (SE, 18.07.2011)
##                   For safety reason the following changes have been made:
##                   - Added Schmitt Trigger I/O for I2C busses
##   Version 6     : (SE, 27.07.2011)
##                   Minor file cosmetics
##
## Comments        :
##   - Net CFG_Dx 0-2 do not have the extension _FSx 0-2
##
## ##### ##
##
## ##### ##
## Section: Miscellaneous
## ##### ##
##
## Define IO Standards
net "CFG_CLK[?]"          iostandard = LVCMOS25;

net "SCL_PL"             iostandard = LVCMOS25;
net "SDA_PL"             iostandard = LVCMOS25;

net "BATTERY_LOW_n"     iostandard = LVCMOS25;

net "WC_n"               iostandard = LVCMOS25;
```

```

net "SCL_CPLD"          iostandard = LVCMOS25;
net "SDA_CPLD"          iostandard = LVCMOS25;

net "E[?]"              iostandard = LVCMOS25;

net "FPGA_RST_n"        iostandard = LVCMOS25;

net "PL_LED2_CTRL"      iostandard = LVCMOS25;

net "USER_SWITCH_CPLD" iostandard = LVCMOS25;

net "INTR"              iostandard = LVCMOS25;

# Define Location Constraints
net "CFG_CLK[0]"        loc = P23;          # Bank 1, 32 MHz On-board Oscillator
net "CFG_CLK[1]"        loc = P27;          # Bank 1, Same as [0]

net "SCL_PL"            loc = P22;          # Bank 1, open drain
net "SDA_PL"            loc = P28;          # Bank 1, open drain

net "BATTERY_LOW_n"     loc = P63;          # Bank 1

net "WC_n"              loc = P55;          # Bank 1

net "SCL_CPLD"          loc = P53;          # Bank 1
net "SDA_CPLD"          loc = P52;          # Bank 1

net "E[0]"              loc = P59;          # Bank 1, EEPROM A8 respectively
net "E[1]"              loc = P58;          # Bank 1, EEPROM A9 respectively
net "E[2]"              loc = P56;          # Bank 1, EEPROM A10 respectively

net "FPGA_RST_n"        loc = P60;          # Bank 1

net "PL_LED2_CTRL"      loc = P44;          # Bank 1

net "USER_SWITCH_CPLD"  loc = P49;          # Bank 1

net "INTR"              loc = P19;          # Bank 1

# Timing Constraints
net "CFG_CLK[?]"        tnm_net = "CFG_CLK";
timespec "TS_CFG_CLK"   = period "CFG_CLK" 32 MHz high 50 %;

# There are two different modes exist for configuration: Master Modes and Slave Modes. In Slave Modes
# the clock is provided by the CPLD. The timing efforts are relaxed since the clock arrives almost to
# the same time at the Platform Flash devices and the FPGA.
# In Master Modes the Clock must pass the CPLD and the configuration data must return within the same
# cycle to the FPGA. Consequently, timing efforts are high.
# Since no the CPLD drivers cannot be selected, the constraint is set as close as possible to the
# hardware limit in order to detect the maximum frequency
timespec "TS_PADS"      = from pads to pads 10 ns;      # Cover direct Connections

# The I2C bus operates at 333.33 kHz respectively 3 us. Hence, the timing constraints are relaxed
#
net "SCL_PL"            offset = out 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing
net "SDA_PL"            offset = out 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing
net "SDA_PL"            offset = in 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing

net "SCL_CPLD"          offset = out 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing
net "SDA_CPLD"          offset = out 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing
net "SDA_CPLD"          offset = in 50 ns after "CFG_CLK[1]"; # Ensure internal expected timing

# Additional Constrains
net "SCL_PL"            open_drain;          # Uses open drain due to pin circuit
net "SCL_PL"            schmitt_trigger;     # For Safty Reason on Control I/O
net "SDA_PL"            open_drain;          # Uses open drain due to pin circuit
net "SDA_PL"            schmitt_trigger;     # For Safty Reason on Control I/O

net "SCL_CPLD"          open_drain;          # Uses open drain due to pin circuit
net "SCL_CPLD"          schmitt_trigger;     # For Safty Reason on Control I/O

```

```

net "SDA_CPLD"                open_drain;      # Uses open drain due to pin circuit
net "SDA_CPLD"                schmitt_trigger; # For Safty Reason on Control I/O

## #####
## Section: Configuration Flash I/II (Shared)
## #####

# Define IO Standards
net "XCF_D[?]"                iostandard = LVCMOS33;

# Define Location Constraints
net "XCF_D[0]"                loc = P76;      # Bank 2, Shared between Flash I/II
net "XCF_D[1]"                loc = P71;      # Bank 2, Shared between Flash I/II
net "XCF_D[2]"                loc = P72;      # Bank 2, Shared between Flash I/II
net "XCF_D[3]"                loc = P73;      # Bank 2, Shared between Flash I/II
net "XCF_D[4]"                loc = P74;      # Bank 2, Shared between Flash I/II
net "XCF_D[5]"                loc = P77;      # Bank 2, Shared between Flash I/II
net "XCF_D[6]"                loc = P70;      # Bank 2, Shared between Flash I/II
net "XCF_D[7]"                loc = P78;      # Bank 2, Shared between Flash I/II

## #####
## Section: Configuration Flash 0
## #####

# Define IO Standards
net "CLK_PF0"                 iostandard = LVCMOS33;

net "CE_PF0_n"                iostandard = LVCMOS33;
net "CEO_PF0_n"                iostandard = LVCMOS33;

net "REV_SEL?_PF0"            iostandard = LVCMOS33;

net "CLKOUT_PF0"              iostandard = LVCMOS33;
net "EN_EXT_SEL_PF0_n"        iostandard = LVCMOS33;

# Define Location Constraints
net "CLK_PF0"                 loc = P79;      # Bank 2

net "CE_PF0_n"                loc = P81;      # Bank 2
net "CEO_PF0_n"                loc = P86;      # Bank 2

net "REV_SELO_PF0"            loc = P68;      # Bank 2
net "REV_SEL1_PF0"            loc = P67;      # Bank 2

net "CLKOUT_PF0"              loc = P66;      # Bank 2
net "EN_EXT_SEL_PF0_n"        loc = P65;      # Bank 2

# Timing Constraints

# Two different paths exist: Xilinx Platform Flash and normal Flash. Xilinx limits the
# maximum frequency for its Platform Flashs to 25ns (serial mode) or 30 ns (parallel).
# For safety reasons the slower path is used below
net "CLK_PF0"                  tnm_net = "CLK_PF0";
timespec "TS_CLK_PF0"          = period "CLK_PF0" 33.33 MHz high 50 %;

## #####
## Section: Configuration Flash 1
## #####

# Define IO Standards
net "CLK_PF1"                 iostandard = LVCMOS33;
net "CE_PF1_n"                iostandard = LVCMOS33;

net "REV_SEL?_PF1"            iostandard = LVCMOS33;

net "CLKOUT_PF1"              iostandard = LVCMOS33;
net "EN_EXT_SEL_PF1_n"        iostandard = LVCMOS33;

```

```

# Define Location Constraints
net "CLK_PF1"                loc = P95;          # Bank 2
net "CE_PF1_n"              loc = P12;          # Bank 2

net "REV_SELO_PF1"          loc = P94;          # Bank 2
net "REV_SEL1_PF1"          loc = P92;          # Bank 2

net "CLKOUT_PF1"            loc = P89;          # Bank 2
net "EN_EXT_SEL_PF1_n"      loc = P91;          # Bank 2

# Timing Constraints

# Two different paths exist: Xilinx Platform Flash and normal Flash. Xilinx limits the
# maximum frequency for its Platform Flashes to 25ns (serial mode) or 30 ns (parallel).
# For safety reasons the slower path is used below
net "CLK_PF1"                tnm_net = "CLK_PF1";
timespec "TS_CLK_PF1"        = period "CLK_PF1" 33.33 MHz high 50 %;

## #####
## Section: MLVDS Telecom Clocks (0-3)
## #####

# Define IO Standarts
net "TC_DE[?]"               iostandard = LVCMOS33;
net "TC_FSEN[?]"            iostandard = LVCMOS33;

# Define Location Constraints
net "TC_DE[0]"               loc = P93;          # Bank 2
net "TC_DE[1]"               loc = P87;          # Bank 2
net "TC_DE[2]"               loc = P82;          # Bank 2
net "TC_DE[3]"               loc = P96;          # Bank 2

net "TC_FSEN[1]"             loc = P85;          # Bank 2
net "TC_FSEN[2]"             loc = P97;          # Bank 2

## #####
## Section: Jitter Attenuator
## #####

# Define IO Standarts
net "J_*"                    iostandard = LVCMOS33;

# Define Location Constraints
net "J_MR"                   loc = P90;          # Bank 2
net "J_PLL_SEL"              loc = P8;          # Bank 2
net "J_BW_SEL"               loc = P9;          # Bank 2

net "J_F_SEL[0]"             loc = P10;         # Bank 2
net "J_F_SEL[1]"             loc = P11;         # Bank 2

## #####
## Section: SPI Flash Interface
## #####

# Define IO Standarts
net "SPI_*"                   iostandard = LVCMOS33;

# Define Location Constraints
net "SPI_MOSI"               loc = P7;          # Bank 2
net "SPI_CS_n"               loc = P80;         # Bank 2
net "SPI_MISO"               loc = P13;         # Bank 2
net "SPI_CLK"                loc = P6;          # Bank 2

```

```

# Additional Constraints
net "SPI_MISO"                pullup;                # Ensure Valid Input Level

## #####
## Section: MMC
## #####

# Define IO Standards
net "PL_RESET_n"             iostandard = LVCMOS33;

# Define Location Constraints
net "PL_RESET_n"             loc = P99;                # Bank 2

config prohibit              = P3;                # Bank 2
config prohibit              = P4;                # Bank 2
config prohibit              = P1;                # Bank 2
config prohibit              = P2;                # Bank 2

config prohibit              = P64;               # Bank 1
config prohibit              = P61;               # Bank 1

## #####
## Section: FPGA Configuration
## #####

# Define IO Standards
net "FPGA_RS[?]"             iostandard = LVCMOS25;

net "CFG_D[?]"               iostandard = LVCMOS25;

net "FCS_n"                  iostandard = LVCMOS25;
net "MOSI"                   iostandard = LVCMOS25;
net "CCLK"                   iostandard = LVCMOS25;
net "INIT_n"                 iostandard = LVCMOS25;
net "DONE"                   iostandard = LVCMOS25;
net "PROGRAM_n"              iostandard = LVCMOS25;

net "HSWAPEN"                iostandard = LVCMOS25;

net "D_IN"                   iostandard = LVCMOS25;
net "M[?]"                   iostandard = LVCMOS25;

# Define Location Constraints
net "FPGA_RS[0]"             loc = P43;                # Bank 1
net "FPGA_RS[1]"             loc = P29;                # Bank 1

net "CFG_D[0]"               loc = P40;                # Bank 1
net "CFG_D[1]"               loc = P41;                # Bank 1
net "CFG_D[2]"               loc = P37;                # Bank 1
net "CFG_D[3]"               loc = P39;                # Bank 1
net "CFG_D[4]"               loc = P32;                # Bank 1
net "CFG_D[5]"               loc = P15;               # Bank 1
net "CFG_D[6]"               loc = P36;                # Bank 1
net "CFG_D[7]"               loc = P35;                # Bank 1

net "FCS_n"                  loc = P30;                # Bank 1
net "MOSI"                   loc = P17;                # Bank 1
net "CCLK"                   loc = P14;                # Bank 1
net "INIT_n"                 loc = P33;                # Bank 1
net "DONE"                   loc = P50;                # Bank 1
net "PROGRAM_n"              loc = P54;                # Bank 1

net "HSWAPEN"                loc = P46;                # Bank 1

net "D_IN"                   loc = P16;                # Bank 1

net "M[0]"                   loc = P34;                # Bank 1
net "M[1]"                   loc = P18;                # Bank 1
net "M[2]"                   loc = P42;                # Bank 1

```

Timing Constraints

Two different paths exist: Xilinx Platform Flash and normal Flash. Xilinx limits the
maximum frequency for its Platform Flashes to 25ns (serial mode) or 30 ns (parallel).
The maximum frequency for the normal SPI Flash is 50 MHz. In accordance to that, the
maximum frequency is set

```
net "CCLK"                tnm_net = "CCLK";  
timespec "TS_CCLK"       = period "CCLK" 50 MHz high 50 %;
```

Additional Constraints

```
net "FPGA_RS[0]"         pullup;           # Ensure Valid Input Level  
net "FPGA_RS[1]"         pullup;           # Ensure Valid Input Level  
  
net "FCS_n"              pullup;           # Recommend in UG191  
net "FCS_n"              schmitt_trigger;  # For Safty Reason on Control I/O  
net "MOSI"               pullup;           # Recommend in UG191  
net "CCLK"               pullup;           # Ensure Valid Input Level  
  
net "DONE"               schmitt_trigger;  # For Safty Reason on Control I/O  
  
net "D_IN"               pullup;           # Recommend in UG191
```

12 Appendix B

This appendix contains the signal to pin assignments for the Virtex-5 FPGA.

```

## #####
##                                     TEWS TECHNOLOGIES                                     ##
## #####
##
## Project Name      : TAMC640 Complete Pinning
## File Name        : tamc640_fpga.ucf
## Target Device    : XC5VxxxxT-xFF1136
## Design Tool      : Xilinx ISE Design Suit Embedded 12.4
## Simulation Tool  : Xilinx ISIM included in Design Tool
##
## Description      : The file lists all FPGA pins that are connected on the TAMC640
##
## Owner           : TEWS TECHNOLOGIES GmbH
##                  Am Bahnhof 7
##                  D-25469 Halstenbek
##
##                  Tel.: +49 / (0)4101 / 4058-0
##                  Fax.: +49 / (0)4101 / 4058-19
##                  e-mail: support@tews.com
##
##                  Copyright (c) 2011
##                  TEWS TECHNOLOGIES GmbH
##
## History          :
##   Version 1     : (SE, 05.07.2010)
##                   Initial Version
##   Version 2     : (NK, 05.11.2010)
##                   - DDR2 and QDR-II Pinout changed,
##                   - DDR2 Address-Bank is now DCI capable,
##                   - DDR2 Addr.- and Conrtol-Signal
##                   - I/O-Standard changed to DCI
##                   - some I/O-Standards corrected
##   Version 3     : (NK, 24.11.2010)
##                   I/O-Standard of DDRx_CKE changed to SSTL18_II
##   Version 4     : (NK, 25.11.2010)
##                   QDR-II SRAM Bank 0:
##                   - Pinout change to improve routing
##                   - I/O-Standard of QDRx_CQ and _CQ_n changed to HSTL_DCI_18
##   Version 5     : (SE, 10.02.2011)
##                   General Revise
##   Version 6     : (SE, 14.02.2011)
##                   - Corrected default I/O standard for DM and DQS (DDR0/1)
##   Version 7     : (SE, 24.02.2011)
##                   - Corrected indexes for DDRx_UDQS_P and DDRx_UDQS_N and DDRx_UDM
##   Version 8     : (SE, 18.04.2011)
##                   - DDR1_BA[2] Updated Bank Assignment
##                   - DDR1_CKE Pin-Location Correction
##   Version 9     : (SE, 04.05.2011)
##                   - Removed VCCAUX config setting (unsupported for Virtex 5 devices)
##                   - Added Pin Location for PL_LED2_1V8
##   Version 10    : (SE, 01.07.2011)
##                   - Corrected SPI Interface Pinning
##   Version 11    : (SE, 29.11.2011)
##                   - Revised sections Miscellaneous and Configuration / User Storage
##                   - Added I/O Standard Comment in Section FMC HB
##
## Comments        : VADJ and VCC_B vary from 0V to 3.3V. Hence the pins with VCCO = VADJ must be
##                  set to an appropriate IO standard that reflects the real VADJ value. These
##                  pins get, as a placeholder, the LVCMOS12 standard, as this will most
##                  likely throw a warning as a reminder.
##                  The constraints for the GTP-transceiver for reference only. Replace
##                  them with the constraints valid for your implementaion (i.e. the .ucf from
##                  the core generator output).
## #####

```

```

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

# The configuration interface is linked to CPLD device. Signals that are dedicated (no dual purpose)
# are prohibited for any application usage
config prohibit = "AE13"; # Configuration FPGA_RS1
config prohibit = "AE12"; # Configuration FPGA_RS0

config prohibit = "AD19"; # Configuration CFG_D0_FS0
config prohibit = "AE19"; # Configuration CFG_D1_FS1
config prohibit = "AE17"; # Configuration CFG_D2_FS2
config prohibit = "AF16"; # Configuration CFG_D3
config prohibit = "AD20"; # Configuration CFG_D4
config prohibit = "AE21"; # Configuration CFG_D5
config prohibit = "AE16"; # Configuration CFG_D6
config prohibit = "AF15"; # Configuration CFG_D7

# For Configuration FCS_N and MOSI refer section Configuration / User Storage

## #####
## Section: Configuration / User Storage
## #####

# SPI access is performed via the FPGA standard interface (refer XAPP1020). User configurable I/O
# pins are prohibited for usage due to safety reasons

# Pin Location constraints
config prohibit = "AE14"; # Configuration FCS_N
config prohibit = "AF14"; # Configuration MOSI

## #####
## Section: Gigabit Ethernet (AMC Lanes 0/1)
## #####

# I/O Standard
net "REF_CLK_01_C_?" iostandard = "LVDS_25";

# Pin Location constraints
net "REF_CLK_01_C_P" loc = "E4"; # BANK 120, On-board generated by SI5338
net "REF_CLK_01_C_N" loc = "D4"; # BANK 120, On-board generated by SI5338

net "V_Tx_P[0]" loc = "B4"; # BANK 120, Port 0
net "V_Tx_N[0]" loc = "B3"; # BANK 120, Port 0
net "V_Rx_P[0]" loc = "A3"; # BANK 120, Port 0
net "V_Rx_N[0]" loc = "A2"; # BANK 120, Port 0

net "V_Tx_P[1]" loc = "E2"; # BANK 120, Port 1
net "V_Tx_N[1]" loc = "D2"; # BANK 120, Port 1
net "V_Rx_P[1]" loc = "D1"; # BANK 120, Port 1
net "V_Rx_N[1]" loc = "C1"; # BANK 120, Port 1

## #####
## Section: Fat Pipe Region (AMC Lanes 4-11)
## #####

# I/O Standard
net "V_REFCLK_47_?" iostandard = "LVDS_25";
net "V_REFCLK_811_?" iostandard = "LVDS_25";

# Pin Location constraints
net "V_REFCLK_47_P" loc = "H4"; # BANK 116, aligned to the first 8-lanes [0-7]
net "V_REFCLK_47_N" loc = "H3"; # BANK 116, on-board generated by SI5338

net "V_REFCLK_811_P" loc = "Y4"; # BANK 114, aligned to the first 8-lanes [8-15]
net "V_REFCLK_811_N" loc = "Y3"; # BANK 114, on-board generated by SI5338

```

```

net "V_Tx_P[4]"          loc = "F2";          # BANK 116, Lane 0
net "V_Tx_N[4]"          loc = "G2";          # BANK 116, Lane 0
net "V_Rx_P[4]"          loc = "G1";          # BANK 116, Lane 0
net "V_Rx_N[4]"          loc = "H1";          # BANK 116, Lane 0

net "V_Tx_P[5]"          loc = "L2";          # BANK 116, Lane 1
net "V_Tx_N[5]"          loc = "K2";          # BANK 116, Lane 1
net "V_Rx_P[5]"          loc = "K1";          # BANK 116, Lane 1
net "V_Rx_N[5]"          loc = "J1";          # BANK 116, Lane 1

net "V_Tx_P[6]"          loc = "M2";          # BANK 112, Lane 2
net "V_Tx_N[6]"          loc = "N2";          # BANK 112, Lane 2
net "V_Rx_P[6]"          loc = "N1";          # BANK 112, Lane 2
net "V_Rx_N[6]"          loc = "P1";          # BANK 112, Lane 2

net "V_Tx_P[7]"          loc = "U2";          # BANK 112, Lane 3
net "V_Tx_N[7]"          loc = "T2";          # BANK 112, Lane 3
net "V_Rx_P[7]"          loc = "T1";          # BANK 112, Lane 3
net "V_Rx_N[7]"          loc = "R1";          # BANK 112, Lane 3

net "V_Tx_P[8]"          loc = "V2";          # BANK 114, Lane 4
net "V_Tx_N[8]"          loc = "W2";          # BANK 114, Lane 4
net "V_Rx_P[8]"          loc = "W1";          # BANK 114, Lane 4
net "V_Rx_N[8]"          loc = "Y1";          # BANK 114, Lane 4

net "V_Tx_P[9]"          loc = "AC2";         # BANK 114, Lane 5
net "V_Tx_N[9]"          loc = "AB2";         # BANK 114, Lane 5
net "V_Rx_P[9]"          loc = "AB1";         # BANK 114, Lane 5
net "V_Rx_N[9]"          loc = "AA1";         # BANK 114, Lane 5

net "V_Tx_P[10]"         loc = "AD2";         # BANK 118, Lane 6
net "V_Tx_N[10]"         loc = "AE2";         # BANK 118, Lane 6
net "V_Rx_P[10]"         loc = "AE1";         # BANK 118, Lane 6
net "V_Rx_N[10]"         loc = "AF1";         # BANK 118, Lane 6

net "V_Tx_P[11]"         loc = "AJ2";         # BANK 118, Lane 7
net "V_Tx_N[11]"         loc = "AH2";         # BANK 118, Lane 7
net "V_Rx_P[11]"         loc = "AH1";         # BANK 118, Lane 7
net "V_Rx_N[11]"         loc = "AG1";         # BANK 118, Lane 7

```

```

## #####
## Section: Extended Options Region (AMC Lanes 12-15)
## #####

```

Define I/O Standards

```

net "Tx_C_[1?]"          iostandard = "LVDS_25"; # 2.5V
net "Rx_C_[1?]"          iostandard = "LVDS_18"; # 1.8V

```

Location Constraints

```

net "Tx_C_P[12]"         loc = "AF20";        # BANK 2
net "Tx_C_N[12]"         loc = "AF21";        # BANK 2
net "Rx_C_P[12]"         loc = "V10";         # BANK 18
net "Rx_C_N[12]"         loc = "V9";          # BANK 18

net "Tx_C_P[13]"         loc = "AF23";        # BANK 2
net "Tx_C_N[13]"         loc = "AG23";        # BANK 2
net "Rx_C_P[13]"         loc = "V8";          # BANK 18
net "Rx_C_N[13]"         loc = "U8";          # BANK 18

net "Tx_C_P[14]"         loc = "AF13";        # BANK 2
net "Tx_C_N[14]"         loc = "AG12";        # BANK 2
net "Rx_C_P[14]"         loc = "W10";         # BANK 18
net "Rx_C_N[14]"         loc = "W9";          # BANK 18

net "Tx_C_P[15]"         loc = "AE22";        # BANK 2
net "Tx_C_N[15]"         loc = "AE23";        # BANK 2
net "Rx_C_P[15]"         loc = "Y11";        # BANK 18
net "Rx_C_N[15]"         loc = "W11";        # BANK 18

```

```

# AMC Port 16 is used for Telecom Clocks

net "Tx_C_P[17]"          loc = "AH15";      # BANK 4
net "Tx_C_N[17]"          loc = "AG15";      # BANK 4
net "Rx_C_P[17]"          loc = "W7";          # BANK 18
net "Rx_C_N[17]"          loc = "V7";          # BANK 18

## #####
## Section: FMC (Miscellaneous)
## #####

# Define I/O Standards
net "FMC_PRESENT_1V8_N"    iostandard = "LVCMOS18"; # 1.8V

net "CLK_DIR_1V8"          iostandard = "LVCMOS18"; # 1.8V

net "SDA_FMC_1V8"          iostandard = "LVCMOS18"; # 1.8V
net "SCL_FMC_1V8"          iostandard = "LVCMOS18"; # 1.8V

net "GBTCLK0_M2C_C_?"      iostandard = "LVDS_25"; # 2.5V
net "CLK?_BIDIR_?"         iostandard = "LVDS_25"; # 2.5V
net "CLK?_M2C_?"           iostandard = "LVDS_25"; # 2.5V

# Location Constraints
net "FMC_PRESENT_1V8_N"    loc = "AA10";      # BANK 21

net "CLK_DIR_1V8"          loc = "AA25";      # BANK 21

net "SDA_FMC_1V8"          loc = "AA9";        # BANK 22
net "SCL_FMC_1V8"          loc = "AA8";        # BANK 22

net "GBTCLK0_M2C_C_P"      loc = "AL5";        # BANK 122
net "GBTCLK0_M2C_C_N"      loc = "AL4";        # BANK 122

net "DP_C2M_P[0]"          loc = "AK2";        # BANK 122, Lane 0
net "DP_C2M_N[0]"          loc = "AL2";        # BANK 122, Lane 0
net "DP_M2C_P[0]"          loc = "AL1";        # BANK 122, Lane 0
net "DP_M2C_N[0]"          loc = "AM1";        # BANK 122, Lane 0

net "DP_C2M_P[1]"          loc = "AN4";        # BANK 122, Lane 1
net "DP_C2M_N[1]"          loc = "AN3";        # BANK 122, Lane 1
net "DP_M2C_P[1]"          loc = "AP3";        # BANK 122, Lane 1
net "DP_M2C_N[1]"          loc = "AP2";        # BANK 122, Lane 1

net "CLK2_BIDIR_P"         loc = "AG18";      # BANK 4
net "CLK2_BIDIR_N"         loc = "AF19";      # BANK 4

net "CLK3_BIDIR_P"         loc = "AH17";      # BANK 4
net "CLK3_BIDIR_N"         loc = "AG16";      # BANK 4

net "CLK0_M2C_P"           loc = "AF18";      # BANK 4
net "CLK0_M2C_N"           loc = "AE18";      # BANK 4

net "CLK1_M2C_P"           loc = "AH18";      # BANK 4
net "CLK1_M2C_N"           loc = "AG17";      # BANK 4

## #####
## Section: FMC LA
## #####

# Define IO Standards
net "LA_*"                  iostandard = "LVDS_12"; # VADJ

# Location Constraints
net "LA_P[0]"               loc = "AH34";      # BANK 13

```

```

net "LA_P[1]"          loc = "AF34";      # BANK 13
net "LA_P[2]"          loc = "AN32";      # BANK 13
net "LA_P[3]"          loc = "AN34";      # BANK 13
net "LA_P[4]"          loc = "AM33";      # BANK 13
net "LA_P[5]"          loc = "AK34";      # BANK 13
net "LA_P[6]"          loc = "AL34";      # BANK 13
net "LA_P[7]"          loc = "AF33";      # BANK 13
net "LA_P[8]"          loc = "AJ32";      # BANK 13
net "LA_P[9]"          loc = "AC34";      # BANK 13
net "LA_P[10]"         loc = "AD32";      # BANK 13
net "LA_P[11]"         loc = "AC33";      # BANK 13
net "LA_P[12]"         loc = "AC32";      # BANK 13
net "LA_P[13]"         loc = "AA34";      # BANK 13
net "LA_P[14]"         loc = "Y33";      # BANK 13
net "LA_P[15]"         loc = "W34";      # BANK 13
net "LA_P[16]"         loc = "V32";      # BANK 13
net "LA_P[17]"         loc = "K33";      # BANK 11
net "LA_P[18]"         loc = "L34";      # BANK 11
net "LA_P[19]"         loc = "U33";      # BANK 11
net "LA_P[20]"         loc = "U32";      # BANK 11
net "LA_P[21]"         loc = "P32";      # BANK 11
net "LA_P[22]"         loc = "T33";      # BANK 11
net "LA_P[23]"         loc = "R33";      # BANK 11
net "LA_P[24]"         loc = "G33";      # BANK 11
net "LA_P[25]"         loc = "J32";      # BANK 11
net "LA_P[26]"         loc = "H34";      # BANK 11
net "LA_P[27]"         loc = "L33";      # BANK 11
net "LA_P[28]"         loc = "F33";      # BANK 11
net "LA_P[29]"         loc = "E32";      # BANK 11
net "LA_P[30]"         loc = "C34";      # BANK 11
net "LA_P[31]"         loc = "C32";      # BANK 11
net "LA_P[32]"         loc = "B32";      # BANK 11
net "LA_P[33]"         loc = "B33";      # BANK 11

net "LA_N[0]"          loc = "AJ34";      # BANK 13
net "LA_N[1]"          loc = "AE34";      # BANK 13
net "LA_N[2]"          loc = "AP32";      # BANK 13
net "LA_N[3]"          loc = "AN33";      # BANK 13
net "LA_N[4]"          loc = "AM32";      # BANK 13
net "LA_N[5]"          loc = "AK33";      # BANK 13
net "LA_N[6]"          loc = "AL33";      # BANK 13
net "LA_N[7]"          loc = "AE33";      # BANK 13
net "LA_N[8]"          loc = "AK32";      # BANK 13
net "LA_N[9]"          loc = "AD34";      # BANK 13
net "LA_N[10]"         loc = "AE32";      # BANK 13
net "LA_N[11]"         loc = "AB33";      # BANK 13
net "LA_N[12]"         loc = "AB32";      # BANK 13
net "LA_N[13]"         loc = "Y34";      # BANK 13
net "LA_N[14]"         loc = "AA33";      # BANK 13
net "LA_N[15]"         loc = "V34";      # BANK 13
net "LA_N[16]"         loc = "V33";      # BANK 13
net "LA_N[17]"         loc = "K32";      # BANK 11
net "LA_N[18]"         loc = "K34";      # BANK 11
net "LA_N[19]"         loc = "T34";      # BANK 11
net "LA_N[20]"         loc = "U31";      # BANK 11
net "LA_N[21]"         loc = "N32";      # BANK 11
net "LA_N[22]"         loc = "R34";      # BANK 11
net "LA_N[23]"         loc = "R32";      # BANK 11
net "LA_N[24]"         loc = "F34";      # BANK 11
net "LA_N[25]"         loc = "H33";      # BANK 11
net "LA_N[26]"         loc = "J34";      # BANK 11
net "LA_N[27]"         loc = "M32";      # BANK 11
net "LA_N[28]"         loc = "E34";      # BANK 11
net "LA_N[29]"         loc = "E33";      # BANK 11
net "LA_N[30]"         loc = "D34";      # BANK 11
net "LA_N[31]"         loc = "D32";      # BANK 11
net "LA_N[32]"         loc = "A33";      # BANK 11
net "LA_N[33]"         loc = "C33";      # BANK 11

```

```

## #####
## Section: FMC HA
## #####

```

```

# Define IO Standards
net "HA_CLK"                iostandard = "LVCMOS12"; # VADJ

net "HA_?[*]"              iostandard = "LVDS_12"; # VADJ

# Location Constraints
net "HA_CLK"                loc = "H17"; # BANK 3

net "HA_P[0]"              loc = "M31"; # BANK 15
net "HA_P[1]"              loc = "P31"; # BANK 15
net "HA_P[2]"              loc = "U27"; # BANK 15
net "HA_P[3]"              loc = "R26"; # BANK 15
net "HA_P[4]"              loc = "U26"; # BANK 15
net "HA_P[5]"              loc = "U25"; # BANK 15
net "HA_P[6]"              loc = "T31"; # BANK 15
net "HA_P[7]"              loc = "T28"; # BANK 15
net "HA_P[8]"              loc = "L30"; # BANK 15
net "HA_P[9]"              loc = "N29"; # BANK 15
net "HA_P[10]"             loc = "J30"; # BANK 15
net "HA_P[11]"             loc = "K31"; # BANK 15
net "HA_P[12]"             loc = "H30"; # BANK 15
net "HA_P[13]"             loc = "H29"; # BANK 15
net "HA_P[14]"             loc = "F31"; # BANK 15
net "HA_P[15]"             loc = "G30"; # BANK 15
net "HA_P[16]"             loc = "E29"; # BANK 15
net "HA_P[17]"             loc = "K17"; # BANK 3
net "HA_P[18]"             loc = "L19"; # BANK 3
net "HA_P[19]"             loc = "H14"; # BANK 3
net "HA_P[20]"             loc = "J14"; # BANK 3
net "HA_P[21]"             loc = "K18"; # BANK 3
net "HA_P[22]"             loc = "J20"; # BANK 3
net "HA_P[23]"             loc = "H19"; # BANK 3

net "HA_N[0]"              loc = "N30"; # BANK 15
net "HA_N[1]"              loc = "P30"; # BANK 15
net "HA_N[2]"              loc = "U28"; # BANK 15
net "HA_N[3]"              loc = "R27"; # BANK 15
net "HA_N[4]"              loc = "T26"; # BANK 15
net "HA_N[5]"              loc = "T25"; # BANK 15
net "HA_N[6]"              loc = "R31"; # BANK 15
net "HA_N[7]"              loc = "T29"; # BANK 15
net "HA_N[8]"              loc = "M30"; # BANK 15
net "HA_N[9]"              loc = "P29"; # BANK 15
net "HA_N[10]"             loc = "J31"; # BANK 15
net "HA_N[11]"             loc = "L31"; # BANK 15
net "HA_N[12]"             loc = "G31"; # BANK 15
net "HA_N[13]"             loc = "J29"; # BANK 15
net "HA_N[14]"             loc = "E31"; # BANK 15
net "HA_N[15]"             loc = "F30"; # BANK 15
net "HA_N[16]"             loc = "F29"; # BANK 15
net "HA_N[17]"             loc = "L18"; # BANK 3
net "HA_N[18]"             loc = "K19"; # BANK 3
net "HA_N[19]"             loc = "H15"; # BANK 3
net "HA_N[20]"             loc = "H13"; # BANK 3
net "HA_N[21]"             loc = "J19"; # BANK 3
net "HA_N[22]"             loc = "J21"; # BANK 3
net "HA_N[23]"             loc = "H20"; # BANK 3

# Timing Constraints
net "HA_CLK"                tnm_net = "HA_CLK";
timespec "TS_HA_CLK"        = period "HA_CLK" 50 MHz high 50 %;

## #####
## Section: FMC HB
## #####

# Define IO Standards
net "HB_?[*]"              iostandard = "LVDS_12"; # VCC_B (FMC provided Power Supply)

```

```

# Location Constraints
net "HB_P[0]"          loc = "G27";          # BANK 19
net "HB_P[1]"          loc = "P26";          # BANK 19
net "HB_P[2]"          loc = "N24";          # BANK 19
net "HB_P[3]"          loc = "R24";          # BANK 19
net "HB_P[4]"          loc = "M25";          # BANK 19
net "HB_P[5]"          loc = "P25";          # BANK 19
net "HB_P[6]"          loc = "H28";          # BANK 19
net "HB_P[7]"          loc = "M28";          # BANK 19
net "HB_P[8]"          loc = "K24";          # BANK 19
net "HB_P[9]"          loc = "L25";          # BANK 19
net "HB_P[10]"         loc = "E28";          # BANK 19
net "HB_P[11]"         loc = "K28";          # BANK 19
net "HB_P[12]"         loc = "H25";          # BANK 19
net "HB_P[13]"         loc = "J24";          # BANK 19
net "HB_P[14]"         loc = "E26";          # BANK 19
net "HB_P[15]"         loc = "G25";          # BANK 19
net "HB_P[16]"         loc = "F25";          # BANK 19
net "HB_P[17]"         loc = "G23";          # BANK 1
net "HB_P[18]"         loc = "K23";          # BANK 1
net "HB_P[19]"         loc = "L15";          # BANK 1
net "HB_P[20]"         loc = "J22";          # BANK 1
net "HB_P[21]"         loc = "L21";          # BANK 1

net "HB_N[0]"          loc = "H27";          # BANK 19
net "HB_N[1]"          loc = "P27";          # BANK 19
net "HB_N[2]"          loc = "P24";          # BANK 19
net "HB_N[3]"          loc = "T24";          # BANK 19
net "HB_N[4]"          loc = "M26";          # BANK 19
net "HB_N[5]"          loc = "N25";          # BANK 19
net "HB_N[6]"          loc = "G28";          # BANK 19
net "HB_N[7]"          loc = "N28";          # BANK 19
net "HB_N[8]"          loc = "L24";          # BANK 19
net "HB_N[9]"          loc = "L26";          # BANK 19
net "HB_N[10]"         loc = "F28";          # BANK 19
net "HB_N[11]"         loc = "L28";          # BANK 19
net "HB_N[12]"         loc = "H24";          # BANK 19
net "HB_N[13]"         loc = "J25";          # BANK 19
net "HB_N[14]"         loc = "E27";          # BANK 19
net "HB_N[15]"         loc = "G26";          # BANK 19
net "HB_N[16]"         loc = "F26";          # BANK 19
net "HB_N[17]"         loc = "H23";          # BANK 1
net "HB_N[18]"         loc = "K22";          # BANK 1
net "HB_N[19]"         loc = "L16";          # BANK 1
net "HB_N[20]"         loc = "K21";          # BANK 1
net "HB_N[21]"         loc = "L20";          # BANK 1

```

```

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

```

Note: Telecom Clocks are controlled by the on-board CPLD device

```

# Define IO Standards
net "TCLK?_?x"          iostandard = "LVCMOS25"; # 2.5V

net "UCLK"              iostandard = "LVCMOS25"; # 2.5V

```

```

# Location Constraints
net "TCLKA_Rx"          loc = "AH20";          # BANK 4
net "TCLKA_Tx"          loc = "AH19";          # BANK 4

net "TCLKB_Rx"          loc = "AH14";          # BANK 4
net "TCLKB_Tx"          loc = "AH13";          # BANK 4

net "TCLKC_Rx"          loc = "AG22";          # BANK 4
net "TCLKC_Tx"          loc = "AH22";          # BANK 4

net "TCLKD_Rx"          loc = "AH12";          # BANK 4
net "TCLKD_Tx"          loc = "AG13";          # BANK 4

```

```
net "UCLK"                loc = "AG21";      # BANK 4, General purpose 50 MHz clock
(single ended)
```

```
# Timing Specification
```

```
net "UCLK"                tnm_net = "UCLK";
timespec "TS_UCLK"        = period "UCLK" 50 MHz high 50 %;
```

```
## #####
## Section: QDR Memory 0
##
## Do not modify the I/O standard of the QDR memory due to board signal integrity
## #####
```

```
# Define IO Standards
```

```
net "QDR0_D*"             iostandard = HSTL_I_18;
net "QDR0_Q*"             iostandard = HSTL_I_DCI_18;

net "QDR0_A*"             iostandard = HSTL_I_18;

net "QDR0_W_n"            iostandard = HSTL_I_18;
net "QDR0_R_n"            iostandard = HSTL_I_18;

net "QDR0_BW_n*"         iostandard = HSTL_I_18;

net "QDR0_CQ*"           iostandard = HSTL_I_DCI_18;
net "QDR0_K*"            iostandard = HSTL_I_18;
```

```
net "QDR0_D[0]"          loc = "M10";      # BANK 20
net "QDR0_D[1]"          loc = "N10";      # BANK 20
net "QDR0_D[2]"          loc = "L9";       # BANK 20
net "QDR0_D[3]"          loc = "L10";      # BANK 20
net "QDR0_D[4]"          loc = "L11";      # BANK 20
net "QDR0_D[5]"          loc = "J9";       # BANK 20
net "QDR0_D[6]"          loc = "H9";       # BANK 20
net "QDR0_D[7]"          loc = "H8";       # BANK 20
net "QDR0_D[8]"          loc = "G8";       # BANK 20
net "QDR0_D[9]"          loc = "B12";      # BANK 20
net "QDR0_D[10]"         loc = "D12";      # BANK 20
net "QDR0_D[11]"         loc = "D11";      # BANK 20
net "QDR0_D[12]"         loc = "E11";      # BANK 20
net "QDR0_D[13]"         loc = "D10";      # BANK 20
net "QDR0_D[14]"         loc = "F11";      # BANK 20
net "QDR0_D[15]"         loc = "F10";      # BANK 20
net "QDR0_D[16]"         loc = "E9";       # BANK 20
net "QDR0_D[17]"         loc = "E8";       # BANK 20
```

```
net "QDR0_Q[0]"          loc = "U7";       # BANK 12
net "QDR0_Q[1]"          loc = "T6";       # BANK 12
net "QDR0_Q[2]"          loc = "T8";       # BANK 12
net "QDR0_Q[3]"          loc = "T9";       # BANK 12
net "QDR0_Q[4]"          loc = "U10";      # BANK 12
net "QDR0_Q[5]"          loc = "T10";      # BANK 12
net "QDR0_Q[6]"          loc = "T11";      # BANK 12
net "QDR0_Q[7]"          loc = "R8";       # BANK 12
net "QDR0_Q[8]"          loc = "R11";      # BANK 12
net "QDR0_Q[9]"          loc = "E7";       # BANK 12
net "QDR0_Q[10]"         loc = "E6";       # BANK 12
net "QDR0_Q[11]"         loc = "G7";       # BANK 12
net "QDR0_Q[12]"         loc = "G6";       # BANK 12
net "QDR0_Q[13]"         loc = "F6";       # BANK 12
net "QDR0_Q[14]"         loc = "F5";       # BANK 12
net "QDR0_Q[15]"         loc = "G5";       # BANK 12
net "QDR0_Q[16]"         loc = "H5";       # BANK 12
net "QDR0_Q[17]"         loc = "J5";       # BANK 12
```

```
net "QDR0_A[0]"          loc = "C12";      # BANK 20
net "QDR0_A[1]"          loc = "H10";      # BANK 20
net "QDR0_A[2]"          loc = "E12";      # BANK 20
net "QDR0_A[3]"          loc = "G11";      # BANK 20
net "QDR0_A[4]"          loc = "K8";       # BANK 20
net "QDR0_A[5]"          loc = "L6";       # BANK 20
```

```

net "QDR0_A[6]"          loc = "M7";          # BANK 20
net "QDR0_A[7]"          loc = "N9";          # BANK 20
net "QDR0_A[8]"          loc = "K7";          # BANK 20
net "QDR0_A[9]"          loc = "M6";          # BANK 12
net "QDR0_A[10]"         loc = "N8";          # BANK 12
net "QDR0_A[11]"         loc = "L8";          # BANK 12
net "QDR0_A[12]"         loc = "M8";          # BANK 12
net "QDR0_A[13]"         loc = "K6";          # BANK 12
net "QDR0_A[14]"         loc = "N7";          # BANK 12
net "QDR0_A[15]"         loc = "P6";          # BANK 12
net "QDR0_A[16]"         loc = "P7";          # BANK 12
net "QDR0_A[17]"         loc = "J10";         # BANK 20
net "QDR0_A[18]"         loc = "B13";         # BANK 20
net "QDR0_A[19]"         loc = "K11";         # BANK 20
net "QDR0_A[20]"         loc = "A13";         # BANK 20

net "QDR0_W_n"           loc = "C13";         # BANK 12
net "QDR0_R_n"           loc = "J11";         # BANK 12

net "QDR0_BW_n[0]"       loc = "G12";         # BANK 20
net "QDR0_BW_n[1]"       loc = "E13";         # BANK 20

net "QDR0_CQ_P[0]"       loc = "R7";          # BANK 12
net "QDR0_CQ_N[0]"       loc = "H7";          # BANK 12

net "QDR0_K_P[0]"         loc = "F13";         # BANK 12
net "QDR0_K_N[0]"         loc = "G13";         # BANK 12

net "QDR0_DOFF_n"        loc = "F9";          # BANK 20

```

```

## #####
## Section: DDR2 Memory 0
##
## Do not modify the I/O standard of the DDR2 memory due to board signal integrity
## #####

```

```

# Define I/O Standards
net "DDR0_DQ[*]"          iostandard = SSTL18_II_DCI;

net "DDR0_A[*]"           iostandard = SSTL18_II_DCI;
net "DDR0_BA[*]"          iostandard = SSTL18_II_DCI;
net "DDR0_RAS_n"          iostandard = SSTL18_II_DCI;
net "DDR0_CAS_n"          iostandard = SSTL18_II_DCI;
net "DDR0_WE_n"           iostandard = SSTL18_II_DCI;

net "DDR0_CS_n[*]"        iostandard = SSTL18_II_DCI;
net "DDR0_ODT[*]"         iostandard = SSTL18_II_DCI;

net "DDR0_CKE[*]"         iostandard = SSTL18_II;

net "DDR0_?DM[*]"         iostandard = SSTL18_II_DCI;

net "DDR0_?DQS_P[*]"      iostandard = DIFF_SSTL18_II_DCI;
net "DDR0_?DQS_N[*]"      iostandard = DIFF_SSTL18_II_DCI;

net "DDR0_CK_P[*]"        iostandard = DIFF_SSTL18_II_DCI;
net "DDR0_CK_N[*]"        iostandard = DIFF_SSTL18_II_DCI;

```

```

# Location Constraints
net "DDR0_DQ[0]"          loc = "AJ7";          # BANK 18
net "DDR0_DQ[1]"          loc = "AC4";          # BANK 18
net "DDR0_DQ[2]"          loc = "AJ6";          # BANK 18
net "DDR0_DQ[3]"          loc = "AD4";          # BANK 18
net "DDR0_DQ[4]"          loc = "AA5";          # BANK 18
net "DDR0_DQ[5]"          loc = "AK6";          # BANK 18
net "DDR0_DQ[6]"          loc = "AB5";          # BANK 18
net "DDR0_DQ[7]"          loc = "AK7";          # BANK 18
net "DDR0_DQ[8]"          loc = "AD7";          # BANK 18
net "DDR0_DQ[9]"          loc = "AB7";          # BANK 18
net "DDR0_DQ[10]"         loc = "AD6";          # BANK 18
net "DDR0_DQ[11]"         loc = "AC5";          # BANK 18

```

```

net "DDR0_DQ[12]"          loc = "AB6";          # BANK 18
net "DDR0_DQ[13]"          loc = "AE6";          # BANK 18
net "DDR0_DQ[14]"          loc = "AA6";          # BANK 18
net "DDR0_DQ[15]"          loc = "AH7";          # BANK 18

net "DDR0_A[0]"            loc = "AJ10";         # BANK 22
net "DDR0_A[1]"            loc = "AF11";         # BANK 22
net "DDR0_A[2]"            loc = "AH10";         # BANK 22
net "DDR0_A[3]"            loc = "AC8";           # BANK 22
net "DDR0_A[4]"            loc = "AK9";           # BANK 22
net "DDR0_A[5]"            loc = "AG10";         # BANK 22
net "DDR0_A[6]"            loc = "AH9";           # BANK 22
net "DDR0_A[7]"            loc = "AC9";           # BANK 22
net "DDR0_A[8]"            loc = "AJ9";           # BANK 22
net "DDR0_A[9]"            loc = "AF9";           # BANK 22
net "DDR0_A[10]"           loc = "AE11";         # BANK 22
net "DDR0_A[11]"           loc = "AH8";           # BANK 22
net "DDR0_A[12]"           loc = "AE8";           # BANK 22

net "DDR0_R[1]"            loc = "AC10";         # BANK 22, Address Pin A[13] for future use
net "DDR0_R[2]"            loc = "AG8";           # BANK 22, Address Pin A[14] for future use
net "DDR0_R[3]"            loc = "AK8";           # BANK 22, Address Pin A[15] for future use

net "DDR0_BA[0]"           loc = "AJ11";         # BANK 22
net "DDR0_BA[1]"           loc = "AK11";         # BANK 22
net "DDR0_BA[2]"           loc = "AG11";         # BANK 22

net "DDR0_RAS_n"           loc = "AM13";         # BANK 22
net "DDR0_CAS_n"           loc = "AL10";         # BANK 22

net "DDR0_WE_n"            loc = "AB8";           # BANK 22

net "DDR0_CS_n[0]"         loc = "AD9";          # BANK 22

net "DDR0_ODT[0]"          loc = "AD10";         # BANK 22

net "DDR0_CKE[0]"          loc = "AD11";         # BANK 22

net "DDR0_LDM[0]"          loc = "AD5";          # BANK 18, DM[0]
net "DDR0_UDM[0]"          loc = "AC7";          # BANK 18, DM[1]

net "DDR0_LDQS_P[0]"       loc = "AG5";          # BANK 18, DQS_P[0]
net "DDR0_LDQS_N[0]"       loc = "AF5";          # BANK 18, DQS_N[0]
net "DDR0_UDQS_P[0]"       loc = "AE7";          # BANK 18, DQS_P[1]
net "DDR0_UDQS_N[0]"       loc = "AF6";          # BANK 18, DQS_N[1]

net "DDR0_CK_P[0]"         loc = "AM12";         # BANK 22
net "DDR0_CK_N[0]"         loc = "AM11";         # BANK 22

## #####
## Section: DDR2 Memory 1
##
## Do not modify the I/O standard of the DDR2 memory due to board signal integrity
## #####

# Define I/O Standards
net "DDR1_DQ[*]"           iostandard = SSTL18_II_DCI;

net "DDR1_A[*]"            iostandard = SSTL18_II_DCI;
net "DDR1_BA[*]"           iostandard = SSTL18_II_DCI;
net "DDR1_RAS_n"           iostandard = SSTL18_II_DCI;
net "DDR1_CAS_n"           iostandard = SSTL18_II_DCI;
net "DDR1_WE_n"            iostandard = SSTL18_II_DCI;
net "DDR1_CS_n[*]"         iostandard = SSTL18_II_DCI;
net "DDR1_ODT[*]"          iostandard = SSTL18_II_DCI;
net "DDR1_CKE[*]"          iostandard = SSTL18_II;

net "DDR1_?DM[*]"          iostandard = SSTL18_II_DCI;

net "DDR1_?DQS_P[*]"        iostandard = DIFF_SSTL18_II_DCI;
net "DDR1_?DQS_N[*]"        iostandard = DIFF_SSTL18_II_DCI;

```

```

net "DDR1_CK_P[*]"          iostandard = DIFF_SSTL18_II_DCI;
net "DDR1_CK_N[*]"          iostandard = DIFF_SSTL18_II_DCI;

# Location Constraints
net "DDR1_DQ[0]"            loc = "AC29";           # BANK 17
net "DDR1_DQ[1]"            loc = "Y27";           # BANK 17
net "DDR1_DQ[2]"            loc = "AD29";           # BANK 17
net "DDR1_DQ[3]"            loc = "AF29";           # BANK 17
net "DDR1_DQ[4]"            loc = "AE29";           # BANK 17
net "DDR1_DQ[5]"            loc = "AD30";           # BANK 17
net "DDR1_DQ[6]"            loc = "Y28";           # BANK 17
net "DDR1_DQ[7]"            loc = "AA29";           # BANK 17
net "DDR1_DQ[8]"            loc = "AA30";           # BANK 17
net "DDR1_DQ[9]"            loc = "V28";           # BANK 17
net "DDR1_DQ[10]"           loc = "W29";           # BANK 17
net "DDR1_DQ[11]"           loc = "V27";           # BANK 17
net "DDR1_DQ[12]"           loc = "W27";           # BANK 17
net "DDR1_DQ[13]"           loc = "Y31";           # BANK 17
net "DDR1_DQ[14]"           loc = "V29";           # BANK 17
net "DDR1_DQ[15]"           loc = "W31";           # BANK 17

net "DDR1_A[0]"             loc = "AK29";           # BANK 21
net "DDR1_A[1]"             loc = "AH27";           # BANK 21
net "DDR1_A[2]"             loc = "AJ27";           # BANK 21
net "DDR1_A[3]"             loc = "AG27";           # BANK 21
net "DDR1_A[4]"             loc = "AK28";           # BANK 21
net "DDR1_A[5]"             loc = "AE28";           # BANK 21
net "DDR1_A[6]"             loc = "AJ26";           # BANK 21
net "DDR1_A[7]"             loc = "AF26";           # BANK 21
net "DDR1_A[8]"             loc = "AK27";           # BANK 21
net "DDR1_A[9]"             loc = "AG25";           # BANK 21
net "DDR1_A[10]"            loc = "AF28";           # BANK 21
net "DDR1_A[11]"            loc = "AE27";           # BANK 21
net "DDR1_A[12]"            loc = "AF25";           # BANK 21

net "DDR1_R[1]"             loc = "AF24";           # BANK 21, Address Pin A[13] for future use
net "DDR1_R[2]"             loc = "AE26";           # BANK 21, Address Pin A[14] for future use
net "DDR1_R[3]"             loc = "AK26";           # BANK 21, Address Pin A[15] for future use

net "DDR1_BA[0]"            loc = "AH28";           # BANK 21
net "DDR1_BA[1]"            loc = "AH29";           # BANK 17
net "DDR1_BA[2]"            loc = "AG28";           # BANK 21

net "DDR1_RAS_n"            loc = "AJ30";           # BANK 21
net "DDR1_CAS_n"            loc = "AJ29";           # BANK 21

net "DDR1_WE_n"             loc = "AH30";           # BANK 21

net "DDR1_CS_n[0]"          loc = "AF30";           # BANK 21

net "DDR1_ODT[0]"           loc = "AF31";           # BANK 21

net "DDR1_CKE[0]"           loc = "AE24";           # BANK 21

net "DDR1_LDM[0]"           loc = "Y29";           # BANK 17, DM[0]
net "DDR1_UDM[0]"           loc = "V30";           # BANK 17, DM[1]

net "DDR1_LDQS_P[0]"        loc = "AB30";           # BANK 17, DQS_P[0]
net "DDR1_LDQS_N[0]"        loc = "AC30";           # BANK 17, DQS_N[0]
net "DDR1_UDQS_P[0]"        loc = "AB31";           # BANK 17, DQS_P[1]
net "DDR1_UDQS_N[0]"        loc = "AA31";           # BANK 17, DQS_N[1]

net "DDR1_CK_P[0]"          loc = "AJ31";           # BANK 21
net "DDR1_CK_N[0]"          loc = "AK31";           # BANK 21

## #####
## Section: Debug Connector
## #####

# Define I/O Standards
net "RX_FPGA"                iostandard = "LVCMOS18"; # 1.8V

```

```

net "TX_FPGA"                iostandard = "LVCMOS18"; # 1.8V

net "USER_SWITCH"           iostandard = "LVCMOS18"; # 1.8V

net "GPIO_FPGA"             iostandard = "LVCMOS18"; # 1.8V

# Location Constraints
net "USER_SWITCH"           loc = "Y24";             # BANK 21
net "GPIO_FPGA"             loc = "AA24";             # BANK 21

net "RX_FPGA"               loc = "AB25";             # BANK 21
net "TX_FPGA"               loc = "AB26";             # BANK 21

## #####
## Section: MMC
## #####

# Define I/O Standards
net "FPGA_RST_n"            iostandard = "LVCMOS25"; # 2.5V

net "SDA_PL_1V8"            iostandard = "LVCMOS18"; # 1.8V
net "SCL_PL_1V8"            iostandard = "LVCMOS18"; # 1.8V

net "PL_LED2_1V8"           iostandard = "LVCMOS18"; # 1.8V

# Location Constraints
net "FPGA_RST_N"            loc = "AG20";             # BANK 4

net "SDA_PL_1V8"            loc = "AB27";             # BANK 21
net "SCL_PL_1V8"            loc = "AC27";             # BANK 21

net "PL_LED2_1V8"           loc = "AB10";             # BANK 22

# Additional Constraints
net "FPGA_RST_n"            pullup;

```