Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

XUF208-256-TQ64 Datasheet

XMOS®

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide you with accurate and comprehensive documentation for the hardware and software components used in this product. To subscribe to receive updates, visit http://www.xmos.com/.

XMOS Ltd. is the owner or licensee of the information in this document and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. XMOS Ltd. makes no representation that the information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS and the XMOS logo are registered trademarks of XMOS Ltd in the United Kingdom and other countries, and may not be used without written permission. Company and product names mentioned in this document are the trademarks or registered trademarks of their respective owners.

1 xCORE Multicore Microcontrollers

The xCORE-200 Series is a comprehensive range of 32-bit multicore microcontrollers that brings the low latency and timing determinism of the xCORE architecture to mainstream embedded applications. Unlike conventional microcontrollers, xCORE multicore microcontrollers execute multiple real-time tasks simultaneously and communicate between tasks using a high speed network. Because xCORE multicore microcontrollers are completely deterministic, you can write software to implement functions that traditionally require dedicated hardware.
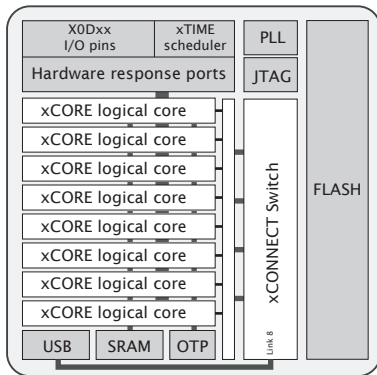

Figure 1:
XUF208-256-TQ64 block diagram

Key features of the XUF208-256-TQ64 include:

▶ **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between five and eight 32-bit xCOREs with highly integrated I/O and on-chip memory.

▶ **Logical cores** Each logical core can execute tasks such as computational code, DSP code, control software (including logic decisions and executing a state machine) or software that handles I/O. Section 6.1

▶ **xTIME scheduler** The xTIME scheduler performs functions similar to an RTOS, in hardware. It services and synchronizes events in a core, so there is no requirement for interrupt handler routines. The xTIME scheduler triggers cores on events generated by hardware resources such as the I/O pins, communication channels and timers. Once triggered, a core runs independently and concurrently to other cores, until it pauses to wait for more events. Section 6.2

▶ **Channels and channel ends** Tasks running on logical cores communicate using channels formed between two channel ends. Data can be passed synchronously or asynchronously between the channel ends assigned to the communicating tasks. Section 6.5

▶ **xCONNECT Switch and Links** Between tiles, channel communications are implemented over a high performance network of xCONNECT Links and routed through a hardware xCONNECT Switch. Section 6.6

▶ **Ports** The I/O pins are connected to the processing cores by Hardware Response ports. The port logic can drive its pins high and low, or it can sample the value on its pins optionally waiting for a particular condition. Section 6.3

▶ **Clock blocks** xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 6.4

▶ **Memory** Each xCORE Tile integrates a bank of SRAM for instructions and data, and a block of one-time programmable (OTP) memory that can be configured for system wide security features. Section 9

▶ **PLL** The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 7

▶ **USB** The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 10

▶ **Flash** The device has a built-in 2MBflash. Section 8

▶ **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 11

1.1 Software

Devices are programmed using C, C++ or xC (C with multicore extensions). XMOS provides tested and proven software libraries, which allow you to quickly add interface and processor functionality such as USB, Ethernet, PWM, graphics driver, and audio EQ to your applications.

1.2 xTIMEcomposer Studio

The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: uniquely in the embedded world, xTIMEcomposer Studio therefore includes a static timing analyzer, cycle-accurate simulator, and high-speed in-circuit instrumentation.

xTIMEcomposer can be driven from either a graphical development environment, or the command line. The tools are supported on Windows, Linux and MacOS X and available at no cost from xmos.com/downloads. Information on using the tools is provided in the xTIMEcomposer User Guide, X3766.

2 XUF208-256-TQ64 Features

▶ **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
- Eight real-time logical cores
- Core share up to 500 MIPS
  — Up to 1000 MIPS in dual issue mode
- Each logical core has:
  — Guaranteed throughput of between $1/5$ and $1/8$ of tile MIPS
  — 16x32bit dedicated registers
- 167 high-density 16/32-bit instructions
  — All have single clock-cycle execution (except for divide)
  — 32x32→64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions

▶ **USB PHY, fully compliant with USB 2.0 specification**

▶ **Programmable I/O**
- 33 general-purpose I/O pins, configurable as input or output
  — Up to 9 x 1bit port, 5 x 4bit port, 3 x 8bit port, 1 x 16bit port
  — 1 xCONNECT link
- Port sampling rates of up to 60 MHz with respect to an external clock
- 32 channel ends for communication with other cores, on or off-chip

▶ **Memory**
- 256KB internal single-cycle SRAM for code and data storage
- 8KB internal OTP for application boot code
- 2MB internal flash for application code and overlays

▶ **Hardware resources**
- 6 clock blocks
- 10 timers
- 4 locks

▶ **JTAG Module for On-Chip Debug**

▶ **Security Features**
- Programming lock disables debug and prevents read-back of memory contents
- AES bootloader ensures secrecy of IP held on external flash memory

▶ **Ambient Temperature Range**
- Commercial qualification: 0 °C to 70 °C
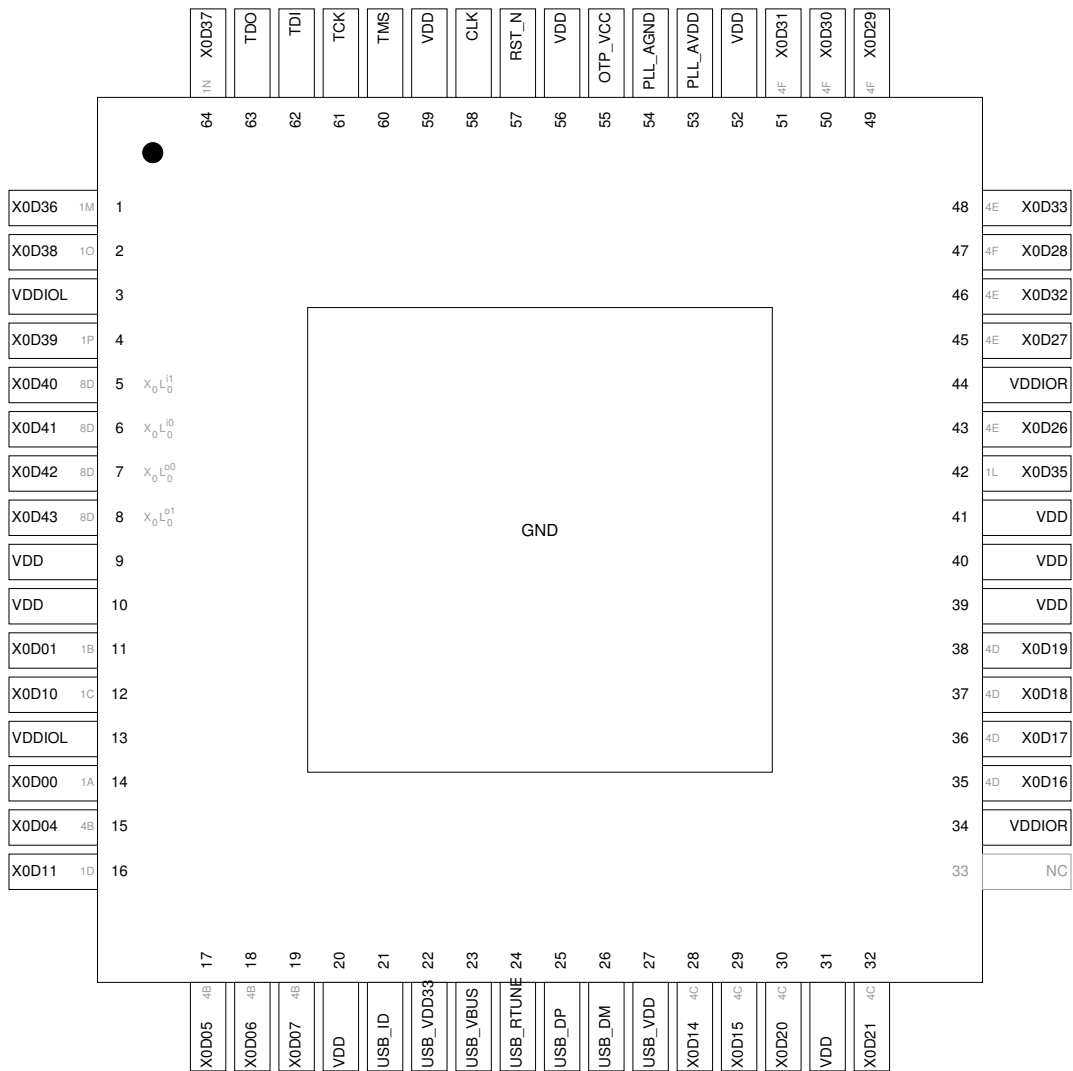- Industrial qualification: -40 °C to 85 °C

▶ **Speed Grade**
- 10: 500 MIPS

▶ **Power Consumption**
- 170 mA (typical)

▶ **64-pin TQFP package 0.5 mm pitch**

3 Pin Configuration

				X0D37	TDO	TDI	TCK	TMS	VDD	CLK	RST_N	VDD	OTP_VCC	PLL_AGND	PLL_AVDD	VDD	X0D31	X0D30	X0D29
				1N													4F	4F	4F
				64	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49

X0D36	1M	1			48	4E	X0D33
X0D38	1O	2			47	4F	X0D28
VDDIOL		3			46	4E	X0D32
X0D39	1P	4			45	4E	X0D27
X0D40	8D	5	$X_0L_0^{i1}$		44		VDDIOR
X0D41	8D	6	$X_0L_0^{i0}$		43	4E	X0D26
X0D42	8D	7	$X_0L_0^{o0}$		42	1L	X0D35
X0D43	8D	8	$X_0L_0^{o1}$		41		VDD
VDD		9		GND	40		VDD
VDD		10			39		VDD
X0D01	1B	11			38	4D	X0D19
X0D10	1C	12			37	4D	X0D18
VDDIOL		13			36	4D	X0D17
X0D00	1A	14			35	4D	X0D16
X0D04	4B	15			34		VDDIOR
X0D11	1D	16			33		NC

		17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
		4B	4B	4B									4C	4C	4C		4C
		X0D05	X0D06	X0D07	VDD	USB_ID	USB_VDD33	USB_VBUS	USB_RTUNE	USB_DP	USB_DM	USB_VDD	X0D14	X0D15	X0D20	VDD	X0D21

4 Signal Description

This section lists the signals and I/O pins available on the XUF208-256-TQ64. The device provides a combination of 1bit, 4bit, 8bit and 16bit ports, as well as wider ports that are fully or partially (gray) bonded out. All pins of a port provide either output or input, but signals in different directions cannot be mapped onto the same port.

Pins may have one or more of the following properties:

▶ PD/PU: The IO pin has a weak pull-down or pull-up resistor. The resistor is enabled during and after reset. Enabling a link or port that uses the pin disables the resistor. Thereafter, the resistor can be enabled or disabled under software control. The resistor is designed to ensure defined logic input state for unconnected pins. It should not be used to pull external circuitry. Note that the resistors are highly non-linear and only a maximum pull current is specified in Section 13.2.

▶ ST: The IO pin has a Schmitt Trigger on its input.

▶ IOL/IOR: The IO pin is powered from VDDIOL, and VDDIOR respectively

Power pins (9)			
Signal	**Function**	**Type**	**Properties**
GND	Digital ground	GND	
OTP_VCC	OTP power supply	PWR	
PLL_AGND	Analog ground for PLL	PWR	
PLL_AVDD	Analog PLL power	PWR	
USB_VDD	Digital tile power	PWR	
USB_VDD33	USB Analog power	PWR	
VDD	Digital tile power	PWR	
VDDIOL	Digital I/O power (left)	PWR	
VDDIOR	Digital I/O power (right)	PWR	

JTAG pins (5)			
Signal	**Function**	**Type**	**Properties**
RST_N	Global reset input	Input	IOL, PU, ST
TCK	Test clock	Input	IOL, PD, ST
TDI	Test data input	Input	IOL, PU
TDO	Test data output	Output	IOL, PD
TMS	Test mode select	Input	IOL, PU

(continued)

Signal	Function				Type	Properties

I/O pins (33)						
Signal	Function				Type	Properties
X0D00	$1A^0$				I/O	IOL, PD
X0D01	$1B^0$				I/O—	IOL, PD
X0D04	$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O—	IOL, PD
X0D05	$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O—	IOL, PD
X0D06	$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O—	IOL, PD
X0D07	$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O—	IOL, PD
X0D10	$1C^0$				I/O—	IOL, PD
X0D11	$1D^0$				I/O	IOL, PD
X0D14	$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	IOR, PD
X0D15	$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	IOR, PD
X0D16	$4D^0$	$8B^2$	$16A^{10}$		I/O	IOR, PD
X0D17	$4D^1$	$8B^3$	$16A^{11}$		I/O	IOR, PD
X0D18	$4D^2$	$8B^4$	$16A^{12}$		I/O	IOR, PD
X0D19	$4D^3$	$8B^5$	$16A^{13}$		I/O	IOR, PD
X0D20	$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X0D21	$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X0D26	$4E^0$	$8C^0$	$16B^0$		I/O	IOR, PD
X0D27	$4E^1$	$8C^1$	$16B^1$		I/O	IOR, PD
X0D28	$4F^0$	$8C^2$	$16B^2$		I/O	IOR, PD
X0D29	$4F^1$	$8C^3$	$16B^3$		I/O	IOR, PD
X0D30	$4F^2$	$8C^4$	$16B^4$		I/O	IOR, PD
X0D31	$4F^3$	$8C^5$	$16B^5$		I/O	IOR, PD
X0D32	$4E^2$	$8C^6$	$16B^6$		I/O	IOR, PD
X0D33	$4E^3$	$8C^7$	$16B^7$		I/O	IOR, PD
X0D35	$1L^0$				I/O	IOR, PD
X0D36	$1M^0$	$8D^0$	$16B^8$		I/O	IOL, PD
X0D37	$1N^0$	$8D^1$	$16B^9$		I/O	IOL, PD
X0D38	$1O^0$	$8D^2$	$16B^{10}$		I/O	IOL, PD
X0D39	$1P^0$	$8D^3$	$16B^{11}$		I/O	IOL, PD
X0D40	$X_0L0_{in}^1$	$8D^4$	$16B^{12}$		I/O	IOL, PD
X0D41	$X_0L0_{in}^0$	$8D^5$	$16B^{13}$		I/O	IOL, PD
X0D42	$X_0L0_{out}^0$	$8D^6$	$16B^{14}$		I/O	IOL, PD
X0D43	$X_0L0_{out}^1$	$8D^7$	$16B^{15}$		I/O	IOL, PD

usb pins (5)			
Signal	Function	Type	Properties
USB_DM	USB Serial Data Inverted	I/O	
USB_DP	USB Serial Data	I/O	
USB_ID	USB Device ID (OTG) - Reserved	I/O	

(continued)

Signal	Function	Type	Properties
USB_RTUNE	USB resistor	I/O	
USB_VBUS	USB Power Detect Pin	I/O	

System pins (1)			
Signal	**Function**	**Type**	**Properties**
CLK	PLL reference clock	Input	IOL, PD, ST

5 Example Application Diagram
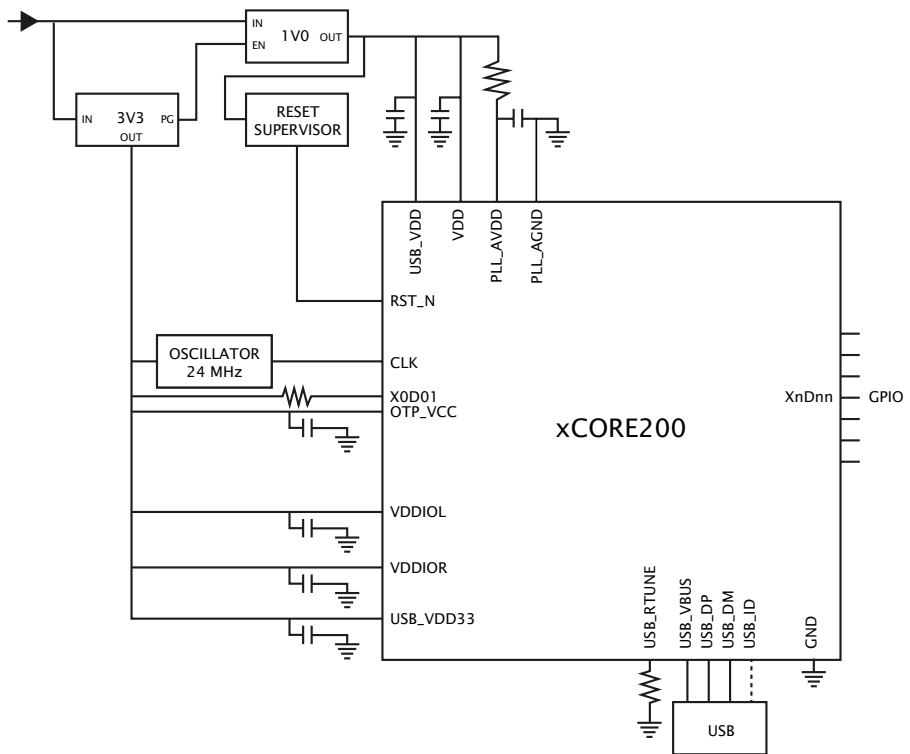
▶ see Section 10 for details on the USB PHY

▶ see Section 12 for details on the power supplies and PCB design

6 Product Overview

The XUF208-256-TQ64 is a powerful device that consists of a single xCORE Tile, which comprises a flexible logical processing cores with tightly integrated I/O and on-chip memory.

6.1 Logical cores

The tile has 8 active logical cores, which issue instructions down a shared five-stage pipeline. Instructions from the active cores are issued round-robin. If up to five logical cores are active, each core is allocated a fifth of the processing cycles. If more than five logical cores are active, each core is allocated at least $1/n$ cycles (for n cores). Figure 3 shows the guaranteed core performance depending on the number of cores used.

Figure 3:
Logical core
performance

Speed grade	MIPS	Frequency	Minimum MIPS per core (for *n* cores)							
			1	2	3	4	5	6	7	8
5	500 MIPS	500 MHz	100	100	100	100	100	83	71	63

There is no way that the performance of a logical core can be reduced below these predicted levels (unless *priority threads* are used: in this case the guaranteed minimum performance is computed based on the number of priority threads as defined in the architecture manual). Because cores may be delayed on I/O, however, their unused processing cycles can be taken by other cores. This means that for more than five logical cores, the performance of each core is often higher than the predicted minimum but cannot be guaranteed.

The logical cores are triggered by events instead of interrupts and run to completion. A logical core can be paused to wait for an event.

6.2 xTIME scheduler

The xTIME scheduler handles the events generated by xCORE Tile resources, such as channel ends, timers and I/O pins. It ensures that all events are serviced and synchronized, without the need for an RTOS. Events that occur at the I/O pins are handled by the Hardware-Response ports and fed directly to the appropriate xCORE Tile. An xCORE Tile can also choose to wait for a specified time to elapse, or for data to become available on a channel.

Tasks do not need to be prioritised as each of them runs on their own logical xCORE. It is possible to share a set of low priority tasks on a single core using cooperative multitasking.

6.3 Hardware Response Ports

Hardware Response ports connect an xCORE tile to one or more physical pins and as such define the interface between hardware attached to the XUF208-256-TQ64, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.
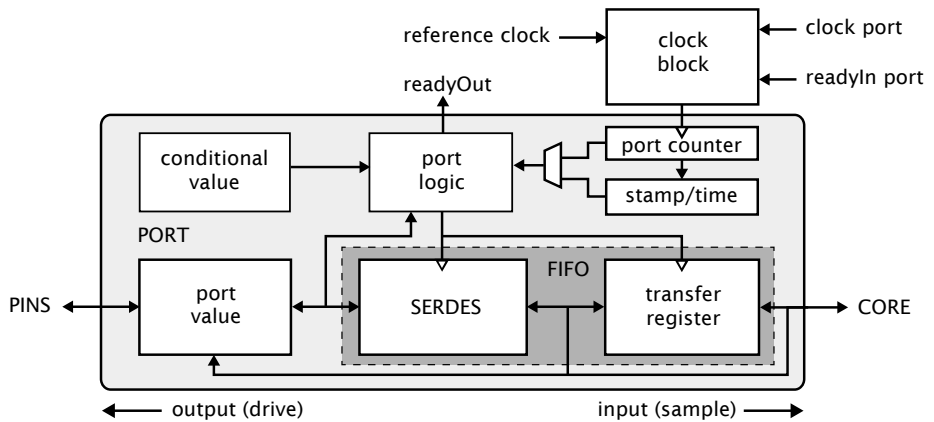
The port logic can drive its pins high or low, or it can sample the value on its pins, optionally waiting for a particular condition. Ports are accessed using dedicated instructions that are executed in a single processor cycle. xCORE-200 IO pins can be used as *open collector* outputs, where signals are driven low if a zero is output, but left high impedance if a one is output. This option is set on a per-port basis.

Data is transferred between the pins and core using a FIFO that comprises a SERDES and transfer register, providing options for serialization and buffered data.

Each port has a 16-bit counter that can be used to control the time at which data is transferred between the port value and transfer register. The counter values can be obtained at any time to find out when data was obtained, or used to delay I/O until some time in the future. The port counter value is automatically saved as a timestamp, that can be used to provide precise control of response times.

The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

6.4 Clock blocks

xCORE devices include a set of programmable clocks called clock blocks that can be used to govern the rate at which ports execute. Each xCORE tile has six clock blocks: the first clock block provides the tile reference clock and runs at a default frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.
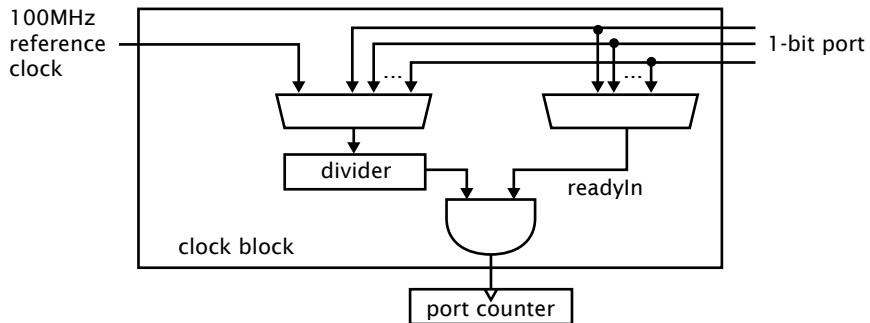
Figure 5:
Clock block
diagram

A clock block can use a 1-bit port as its clock source allowing external application clocks to be used to drive the input and output interfaces. xCORE-200 clock blocks optionally divide the clock input from a 1-bit port.

In many cases I/O signals are accompanied by strobing signals. The xCORE ports can input and interpret strobe (known as readyIn and readyOut) signals generated by external sources, and ports can generate strobe signals to accompany output data.

On reset, each port is connected to clock block 0, which runs from the xCORE Tile reference clock.

6.5   Channels and Channel Ends

Logical cores communicate using point-to-point connections, formed between two channel ends. A channel-end is a resource on an xCORE tile, that is allocated by the program. Each channel-end has a unique system-wide identifier that comprises a unique number and their tile identifier. Data is transmitted to a channel-end by an output-instruction; and the other side executes an input-instruction. Data can be passed synchronously or asynchronously between the channel ends.

6.6   xCONNECT Switch and Links

XMOS devices provide a scalable architecture, where multiple xCORE devices can be connected together to form one system. Each xCORE device has an xCONNECT interconnect that provides a communication infrastructure for all tasks that run on the various xCORE tiles on the system.

The interconnect relies on a collection of switches and XMOS links. Each xCORE device has an on-chip switch that can set up circuits or route data. The switches are connected by xConnect Links. An XMOS link provides a physical connection between two switches. The switch has a routing algorithm that supports many different topologies, including lines, meshes, trees, and hypercubes.

The links operate in either 2 wires per direction or 5 wires per direction mode, depending on the amount of bandwidth required. Circuit switched, streaming
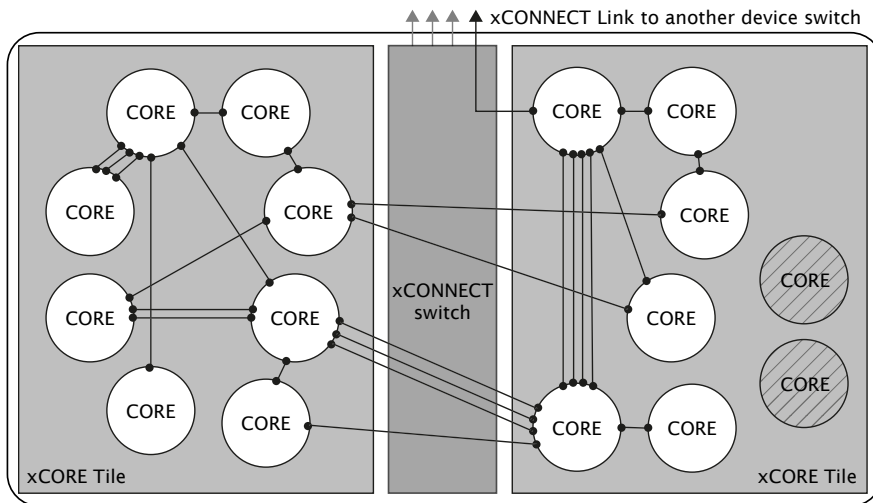
and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between xCORE Tiles (up to 250 MBit/s), but each stream requires a single link to be reserved between switches on two tiles. All packet communications can be multiplexed onto a single link.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-UF Link Performance and Design Guide, X2999.

7 PLL

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock. The initial PLL multiplication value is shown in Figure 7:

Figure 7:
The initial PLL
multiplier
values

Oscillator Frequency	Tile Boot Frequency	PLL Ratio	PLL settings		
			OD	F	R
9-25 MHz	144-400 MHz	16	1	63	0

Figure 7 also lists the values of OD, F and R, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

OD, F and R must be chosen so that $0 \leq R \leq 63$, $0 \leq F \leq 4095$, $0 \leq OD \leq 7$, and $260MHz \leq F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \leq 1.3GHz$. The OD, F, and R values can be modified by writing to the digital node PLL configuration register.

If the USB PHY is used, then either a 24 MHz or 12 MHz oscillator must be used.

If a different tile frequency is required (eg, 500 MHz), then the PLL must be reprogrammed after boot to provide the required tile frequency. The XMOS tools perform this operation by default. Further details on configuring the clock can be found in the xCORE-200 Clock Frequency Control document.

8   Boot Procedure

The device is kept in reset by driving RST_N low. When in reset, all GPIO pins have a pull-down enabled. The processor must be held in reset until VDDIOL is in spec for at least 1 ms. When the device is taken out of reset by releasing RST_N the processor starts its internal reset process. After 15-150 μs (depending on the input clock) the processor boots.

The device boots from a QSPI flash (IS25LQ016B) that is embedded in the device. The QSPI flash is connected to the ports on Tile 0 as shown in Figure 8. An external 1K resistor must connect X0D01 to VDDIOL. X0D10 should ideally not be connected. If X0D10 is connected, then a 150 ohm series resistor close to the device is recommended. X0D04..X0D07 should be not connected.
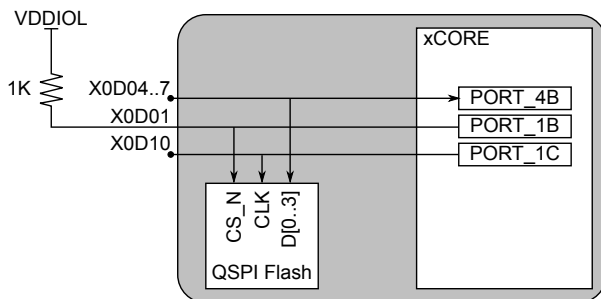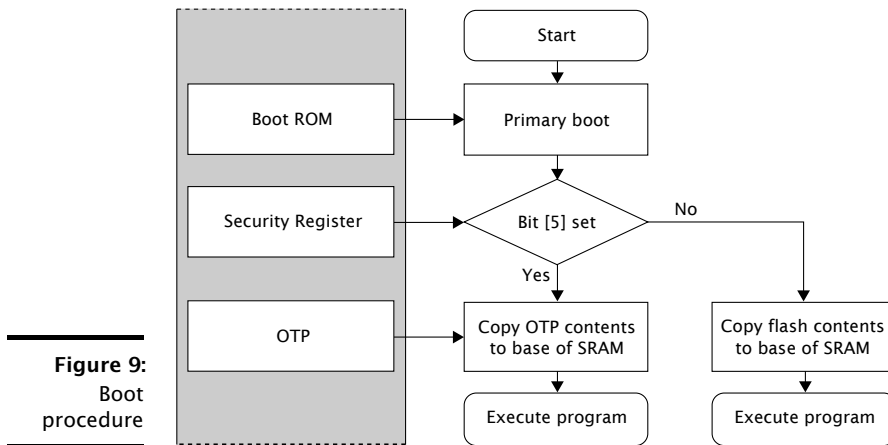

Figure 8:
QSPI port
connectivity

The xCORE Tile boot procedure is illustrated in Figure 9. If bit 5 of the security register (*see* §9.1) is set, the device boots from OTP. Otherwise, the device boots from the internal flash.

The boot image has the following format:

▶ A 32-bit program size s in words.

▶ Program consisting of $s \times 4$ bytes.

▶ A 32-bit CRC, or the value 0x0D15AB1E to indicate that no CRC check should be performed.

The program size and CRC are stored least significant byte first. The program is loaded into the lowest memory address of RAM, and the program is started from that address. The CRC is calculated over the byte stream represented by the program size and the program itself. The polynomial used is 0xEDB88320 (IEEE

802.3); the CRC register is initialized with 0xFFFFFFFF and the residue is inverted to produce the CRC.

8.1 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 10 provide a strong level of protection and are sufficient for providing strong IP security.

9 Memory

9.1 OTP

The xCORE Tile integrates 8 KB one-time programmable (OTP) memory along with a security register that configures system wide security features. The OTP holds data in four sectors each containing 512 rows of 32 bits which can be used to implement secure bootloaders and store encryption keys. Data for the security register is loaded from the OTP on power up. All additional data in OTP is copied from the OTP to SRAM and executed first on the processor.

The OTP memory is programmed using three special I/O ports: the OTP address port is a 16-bit port with resource ID 0x100200, the OTP data is written via a 32-bit port with resource ID 0x200100, and the OTP control is on a 16-bit port with ID 0x100300. Programming is performed through `libotp` and `xburn`.

9.2 SRAM

The xCORE Tile integrates a single 256KB SRAM bank for both instructions and data. All internal memory is 32 bits wide, and instructions are either 16-bit or 32-bit. Byte (8-bit), half-word (16-bit) or word (32-bit) accesses are supported and

Feature	Bit	Description
Disable JTAG	0	The JTAG interface is disabled, making it impossible for the tile state or memory content to be accessed via the JTAG interface.
Disable Link access	1	Other tiles are forbidden access to the processor state via the system switch. Disabling both JTAG and Link access transforms an xCORE Tile into a "secure island" with other tiles free for non-secure user application code.
Secure Boot	5	The xCORE Tile is forced to boot from address 0 of the OTP, allowing the xCORE Tile boot ROM to be bypassed (*see* §8).
Redundant rows	7	Enables redundant rows in OTP.
Sector Lock 0	8	Disable programming of OTP sector 0.
Sector Lock 1	9	Disable programming of OTP sector 1.
Sector Lock 2	10	Disable programming of OTP sector 2.
Sector Lock 3	11	Disable programming of OTP sector 3.
OTP Master Lock	12	Disable OTP programming completely: disables updates to all sectors and security register.
Disable JTAG-OTP	13	Disable all (read & write) access from the JTAG interface to this OTP.
	21..15	General purpose software accessable security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 10:
Security
register
features

are executed within one tile clock cycle. There is no dedicated external memory interface, although data memory can be expanded through appropriate use of the ports.

10   USB PHY

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, XUD, is provided to implement *USB-device* functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 11. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles. Two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.
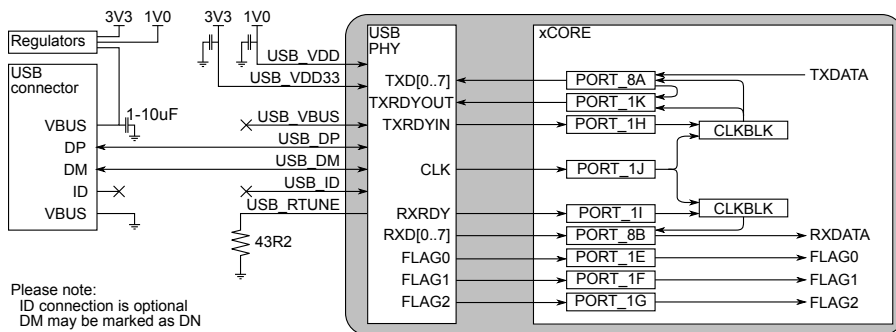
Figure 11:
Bus powered
USB-device

An external resistor of 43.2 ohm (1% tolerance) should connect USB_RTUNE to ground, as close as possible to the device.

10.1 USB VBUS

USB_VBUS need not be connected if the device is wholly powered by USB, and the device is used to implement a *USB-device*.

If you use the USB PHY to design a self-powered *USB-device*, then the device must be able detect the presence of VBus on the USB connector (so the device can disconnect its pull-up resistors from D+/D- to ensure the device does not have any voltage on the D+/D- pins when VBus is not present, "USB Back Voltage Test"). This requires USB_VBUS to be connected to the VBUS pin of the USB connector as is shown in Figure 12.
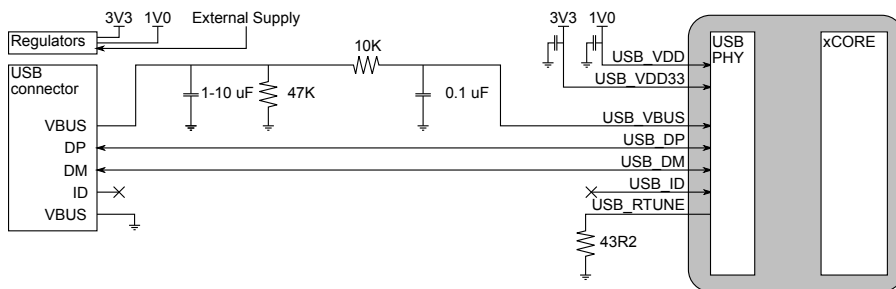

Figure 12:
Self powered
USB-device

When connecting a USB cable to the device it is possible an overvoltage transient will be present on VBus due to the inductance of the USB cable combined with the required input capacitor on VBus. The circuit in Figure 12 ensures that the transient does not damage the device. The 10k series resistor and 0.1uF capacitor ensure than any input transient is filtered and does not reach the device. The 47k resistor to ground is a bleeder resistor to discharge the input capacitor when VBus is not present. The 1-10uF input capacitor is required as part of the USB specification. A typical value would be 2.2uF to ensure the 1uF minimum requirement is met even under voltage bias conditions.

In any case, extra components (such as a ferrite bead and diodes) may be required for EMC compliance and ESD protection. Different wiring is required for USB-host and USB-OTG.

10.2 Logical Core Requirements

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

11 JTAG

The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory.
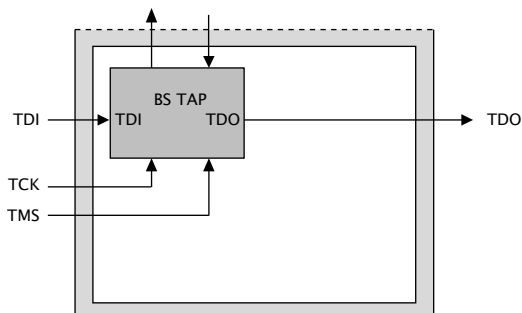

Figure 13:
JTAG chain
structure

The JTAG chain structure is illustrated in Figure 13. It comprises a single 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. It has a 4-bit IR and 32-bit DR. It also provides access to a chip TAP that in turn can access the xCORE Tile for loading code and debugging.

The JTAG module can be reset by holding TMS high for five clock cycles.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 14.

Figure 14:
IDCODE
return value

Bit31				Device Identification Register																										Bit0
Version				Part Number																Manufacturer Identity										1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1
	0				0				0				0				6				6				3				3		

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 15. The OTP User ID field is read from bits [22:31] of the security register , *see* §9.1 (all zero on unprogrammed devices).

Bit31													Usercode Register																	Bit0
OTP User ID										Unused					Silicon Revision																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0			0			0			2			8			0			0			0										

12 Board Integration

The device has the following power supply pins:

▶ VDD pins for the xCORE Tile, including a USB_VDD pin that powers the USB PHY

▶ VDDIO pins for the I/O lines. Separate I/O supplies are provided for the left, and right side of the package; different I/O voltages may be supplied on those. The signal description (Section 4) specifies which I/O is powered from which power-supply. VDDIOL must be a 3.3V supply.

▶ PLL_AVDD pins for the PLL

▶ OTP_VCC pins for the OTP

▶ A USB_VDD33 pin for the analogue supply to the USB-PHY

Several pins of each type are provided to minimize the effect of inductance within the package, all of which must be connected. The power supplies must be brought up monotonically and input voltages must not exceed specification at any time.

The VDD supply must ramp from 0 V to its final value within 10 ms to ensure correct startup.

The VDDIO and OTP_VCC supply must ramp to its final value before VDD reaches 0.4 V.

The PLL_AVDD supply should be separated from the other noisier supplies on the board. The PLL requires a very clean power supply, and a low pass filter (for example, a 4.7 Ω resistor and 100 nF multi-layer ceramic capacitor) is recommended on this pin.

The following ground pins are provided:

▶ PLL_AGND for PLL_AVDD

▶ GND for all other supplies

All ground pins must be connected directly to the board ground.

The VDD and VDDIO supplies should be decoupled close to the chip by several 100 nF low inductance multi-layer ceramic capacitors between the supplies and GND (for example, 100nF 0402 for each supply pin). The ground side of the decoupling

capacitors should have as short a path back to the GND pins as possible. A bulk decoupling capacitor of at least 10 uF should be placed on each of these supplies.

RST_N is an active-low asynchronous-assertion global reset signal. Following a reset, the PLL re-establishes lock after which the device boots up according to the boot mode (*see* §8). RST_N and must be asserted low during and after power up for 100 ns.

12.1 USB connections

USB_VBUS should be connected to the VBUS pin of the USB connector. A 2.2 uF capacitor to ground is required on the VBUS pin. A ferrite bead may be used to reduce HF noise.

For self-powered systems, a bleeder resistor may be required to stop VBUS from floating when no USB cable is attached.

USB_DP and USB_DN should be connected to the USB connector. USB_ID does not need to be connected.

12.2 USB signal routing and placement

The USB_DP and USB_DN lines are the positive and negative data polarities of a high speed USB signal respectively. Their high-speed differential nature implies that they must be coupled and properly isolated. The board design must ensure that the board traces for USB_DP and USB_DN are tightly matched. In addition, according to the USB 2.0 specification, the USB_DP and USB_DN differential impedance must be 90 Ω.

Figure 16: USB trace separation showing a low speed signal, two differential pairs and a high-speed clock
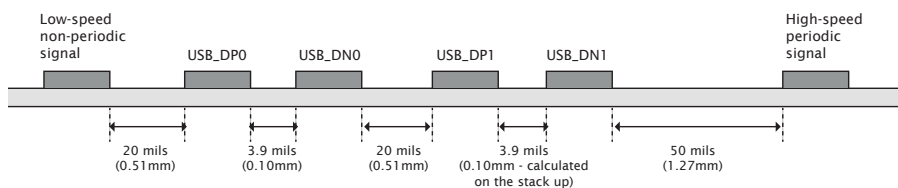
12.2.1 General routing and placement guidelines

The following guidelines will help to avoid signal quality and EMI problems on high speed USB designs. They relate to a four-layer (Signal, GND, Power, Signal) PCB.

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS2-UF8A-256-TQ64 are on the top layer) closest to GND. Reference planes should be below the transmission lines in order to maintain control of the trace impedance.
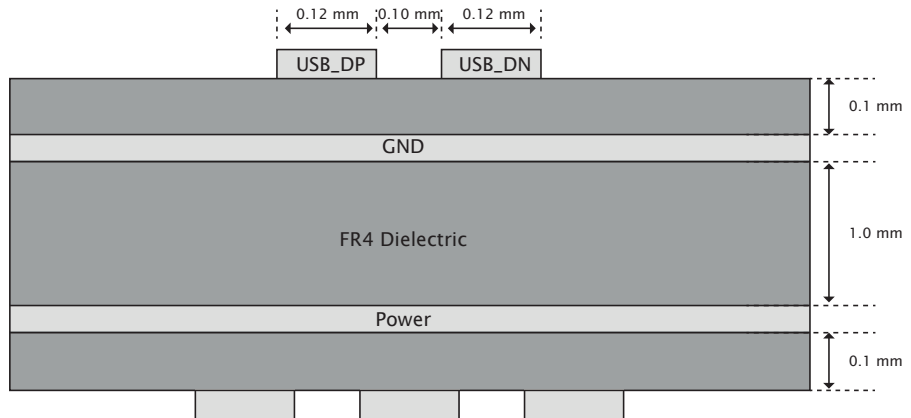
Figure 17:
Example USB
board stack

We recommend that the high-speed clock and high-speed USB differential pairs are routed first before any other routing. When routing high speed USB signals, the following guidelines should be followed:

▶ High speed differential pairs should be routed together.

▶ High-speed USB signal pair traces should be trace-length matched. Maximum trace-length mismatch should be no greater than 4mm.

▶ Ensure that high speed signals (clocks, USB differential pairs) are routed as far away from off-board connectors as possible.

▶ High-speed clock and periodic signal traces that run parallel should be at least 1.27mm away from USB_DP/USB_DN (see Figure 16).

▶ Low-speed and non-periodic signal traces that run parallel should be at least 0.5mm away from USB_DP/USB_DN (see Figure 16).

▶ Route high speed USB signals on the top of the PCB wherever possible.

▶ Route high speed USB traces over continuous power planes, with no breaks. If a trade-off must be made, changing signal layers is preferable to crossing plane splits.

▶ Follow the $20 \times h$ rule; keep traces $20 \times h$ (the height above the power plane) away from the edge of the power plane.

▶ Use a minimum of vias in high speed USB traces.

▶ Avoid corners in the trace. Where necessary, rather than turning through a 90 degree angle, use two 45 degree turns or an arc.

▶ DO NOT route USB traces near clock sources, clocked circuits or magnetic devices.

▶ Avoid stubs on high speed USB signals.

12.3 Land patterns and solder stencils

The package is a 64 pin Thin Quad Flat Package (TQFP) with exposed ground paddle/heat slug on a 0.5mm pitch.

The land patterns and solder stencils will depend on the PCB manufacturing process. We recommend you design them with using the IPC specifications *"Generic Requirements for Surface Mount Design and Land Pattern Standards"* IPC-7351B. This standard aims to achieve desired targets of heel, toe and side fillets for solder-joints. The mechanical drawings in Section 14 specify the dimensions and tolerances.

12.4 Ground and Thermal Vias

Vias under the heat slug into the ground plane of the PCB are recommended for a low inductance ground connection and good thermal performance. Typical designs could use 16 vias in a 4 x 4 grid, equally spaced across the heat slug.

12.5 Moisture Sensitivity

XMOS devices are, like all semiconductor devices, susceptible to moisture absorption. When removed from the sealed packaging, the devices slowly absorb moisture from the surrounding environment. If the level of moisture present in the device is too high during reflow, damage can occur due to the increased internal vapour pressure of moisture. Example damage can include bond wire damage, die lifting, internal or external package cracks and/or delamination.

All XMOS devices are Moisture Sensitivity Level (MSL) 3 - devices have a shelf life of 168 hours between removal from the packaging and reflow, provided they are stored below 30C and 60% RH. If devices have exceeded these values or an included moisture indicator card shows excessive levels of moisture, then the parts should be baked as appropriate before use. This is based on information from *Joint IPC/JEDEC Standard For Moisture/Reflow Sensitivity Classification For Nonhermetic Solid State Surface-Mount Devices* J-STD-020 Revision D.

13 DC and Switching Characteristics

13.1 Operating Conditions

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDD	Tile DC supply voltage	0.95	1.00	1.05	V	
VDDIOL	I/O supply voltage	3.135	3.30	3.465	V	
VDDIOR	I/O supply voltage	3.135	3.30	3.465	V	
USB_VDD	USB tile DC supply voltage	0.95	1.00	1.05	V	
VDD33	Peripheral supply	3.135	3.30	3.465	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Ta	Ambient operating temperature (Commercial)	0		70	℃	
	Ambient operating temperature (Industrial)	-40		85	℃	
Tj	Junction temperature			125	℃	
Tstg	Storage temperature	-65		150	℃	

13.2 DC Characteristics, VDDIO=3V3

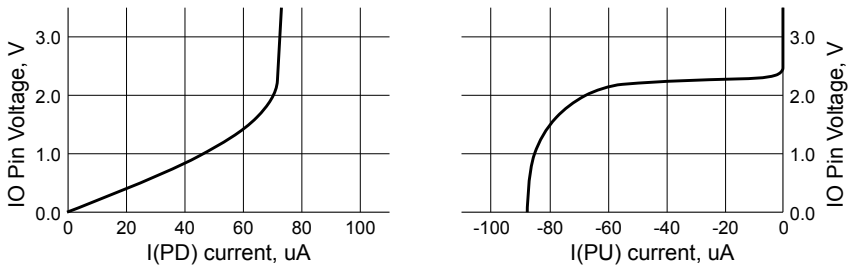| Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	A
V(IL)	Input low voltage	-0.30		0.70	V	A
V(OH)	Output high voltage	2.20			V	B, C
V(OL)	Output low voltage			0.40	V	B, C
I(PU)	Internal pull-up current (Vin=0V)	-100			µA	D
I(PD)	Internal pull-down current (Vin=3.3V)			100	µA	D
I(LC)	Input leakage current	-10		10	µA	

A All pins except power supply pins.

B All general-purpose I/Os are nominal 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry. In order to pull the pin to the opposite state, a 4K7 resistor is recommended to overome the internal pull current.

Figure 20:
Typical internal pull-down and pull-up currents

13.3 ESD Stress Voltage

Figure 21:
ESD stress
voltage

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
HBM	Human body model	-2.00		2.00	KV	
CDM	Charged Device Model	-500		500	V	

13.4 Reset Timing

Figure 22:
Reset timing

Symbol	Parameters	MIN	TYP	MAX	UNITS	Notes
T(RST)	Reset pulse width	5			µs	
T(INIT)	Initialization time			150	µs	A

A Shows the time taken to start booting after RST_N has gone high.