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

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

XL216-512-TQ128 Datasheet

2017/02/02 Document Number: X006989,

Table of Contents

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.

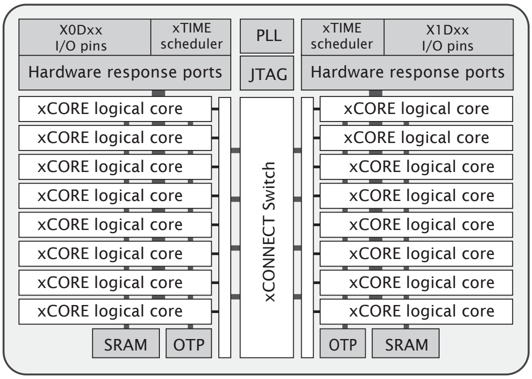


Figure 1: XL216-512-TQ128 block diagram

Key features of the XL216-512-TQ128 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
- **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 10

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

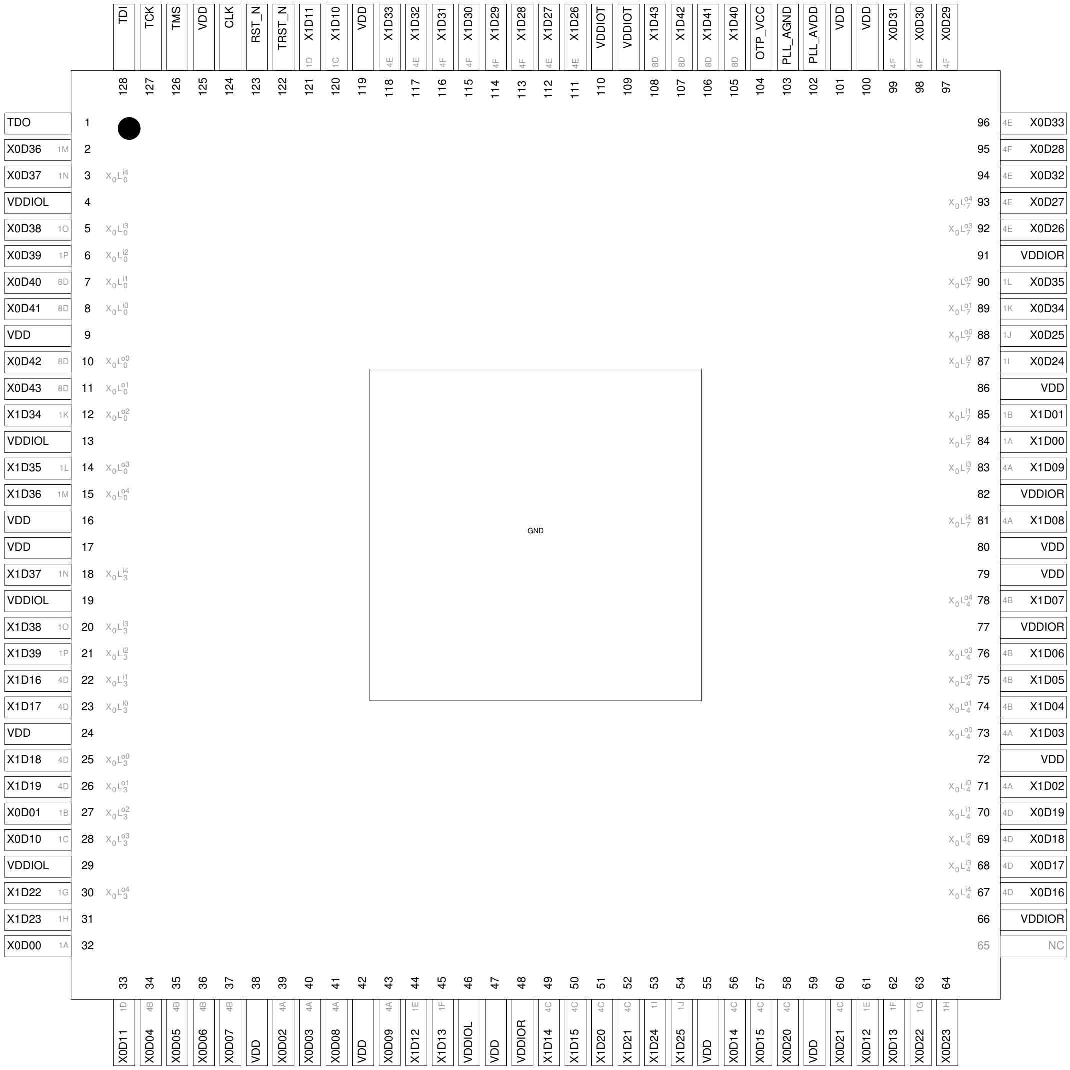
The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: 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 XL216-512-TQ128 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - 16 real-time logical cores on 2 xCORE tiles
  - Cores share up to 1000 MIPS
    - — Up to 2000 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
- **Programmable I/O**
  - 88 general-purpose I/O pins, configurable as input or output
    - — Up to 32 x 1bit port, 12 x 4bit port, 8 x 8bit port, 4 x 16bit port
    - — 4 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 64 channel endss (32 per tile) for communication with other cores, on or off-chip
- **Memory**
  - 512KB internal single-cycle SRAM (max 256KB per tile) for code and data storage
  - 16KB internal OTP (max 8KB per tile) for application boot code
- **Hardware resources**
  - 12 clock blocks (6 per tile)
  - 20 timers (10 per tile)
  - 8 locks (4 per tile)
- **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**
  - 20: 1000 MIPS
- **Power Consumption**
  - 570 mA (typical)
- **128-pin TQFP package 0.4 mm pitch**

3 Pin Configuration



4 Signal Description

This section lists the signals and I/O pins available on the XL216-512-TQ128. 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 12.2.
- ST: The IO pin has a Schmitt Trigger on its input.
- IOL/IOT/IOR: The IO pin is powered from VDDIOL, VDDIOT, and VDDIOR respectively

Power pins (8)			
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	
VDD	Digital tile power	PWR	
VDDIOL	Digital I/O power (left)	PWR	
VDDIOR	Digital I/O power (right)	PWR	
VDDIOT	Digital I/O power (top)	PWR	

JTAG pins (6)			
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
TRST_N	Test reset input	Input	IOL, PU, ST

I/O pins (88)								
Signal	**Function**						**Type**	**Properties**
X0D00		$1A^0$					I/O	IOL, PD
X0D01	$X_0L3^2_{out}$	$1B^0$					I/O	IOL, PD
X0D02			$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	IOL, PD
X0D03			$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	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
X0D08			$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	IOL, PD
X0D09			$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	IOL, PD
X0D10	$X_0L3^3_{out}$	$1C^0$					I/O	IOL, PD
X0D11		$1D^0$					I/O	IOL, PD
X0D12		$1E^0$					I/O	IOR, PD
X0D13		$1F^0$					I/O	IOR, 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	$X_0L4^4_{in}$		$4D^0$	$8B^2$	$16A^{10}$		I/O	IOR, PD
X0D17	$X_0L4^3_{in}$		$4D^1$	$8B^3$	$16A^{11}$		I/O	IOR, PD
X0D18	$X_0L4^2_{in}$		$4D^2$	$8B^4$	$16A^{12}$		I/O	IOR, PD
X0D19	$X_0L4^1_{in}$		$4D^3$	$8B^5$	$16A^{13}$		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
X0D22		$1G^0$					I/O	IOR, PD
X0D23		$1H^0$					I/O	IOR, PD
X0D24	$X_0L7^0_{in}$	$1I^0$					I/O	IOR, PD
X0D25	$X_0L7^0_{out}$	$1J^0$					I/O	IOR, PD
X0D26	$X_0L7^3_{out}$		$4E^0$	$8C^0$	$16B^0$		I/O	IOR, PD
X0D27	$X_0L7^4_{out}$		$4E^1$	$8C^1$	$16B^1$		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
X0D34	$X_0L7^1_{out}$	$1K^0$					I/O	IOR, PD
X0D35	$X_0L7^2_{out}$	$1L^0$					I/O	IOR, PD
X0D36		$1M^0$		$8D^0$	$16B^8$		I/O	IOL, PD
X0D37	$X_0L0^4_{in}$	$1N^0$		$8D^1$	$16B^9$		I/O	IOL, PD
X0D38	$X_0L0^3_{in}$	$1O^0$		$8D^2$	$16B^{10}$		I/O	IOL, PD
X0D39	$X_0L0^2_{in}$	$1P^0$		$8D^3$	$16B^{11}$		I/O	IOL, PD
X0D40	$X_0L0^1_{in}$			$8D^4$	$16B^{12}$		I/O	IOL, PD

(continued)

Signal	Function						Type	Properties
X0D41	$X_0L0^0_{in}$			$8D^5$	$16B^{13}$		I/O	IOL, PD
X0D42	$X_0L0^0_{out}$			$8D^6$	$16B^{14}$		I/O	IOL, PD
X0D43	$X_0L0^1_{out}$			$8D^7$	$16B^{15}$		I/O	IOL, PD
X1D00	$X_0L7^2_{in}$	$1A^0$					I/O	IOR, PD
X1D01	$X_0L7^1_{in}$	$1B^0$					I/O	IOR, PD
X1D02	$X_0L4^0_{in}$		$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	IOR, PD
X1D03	$X_0L4^0_{out}$		$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	I/O	IOR, PD
X1D04	$X_0L4^1_{out}$		$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O	IOR, PD
X1D05	$X_0L4^2_{out}$		$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O	IOR, PD
X1D06	$X_0L4^3_{out}$		$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O	IOR, PD
X1D07	$X_0L4^4_{out}$		$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O	IOR, PD
X1D08	$X_0L7^4_{in}$		$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	IOR, PD
X1D09	$X_0L7^3_{in}$		$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	IOR, PD
X1D10		$1C^0$					I/O	IOT, PD
X1D11		$1D^0$					I/O	IOT, PD
X1D12		$1E^0$					I/O	IOL, PD
X1D13		$1F^0$					I/O	IOL, PD
X1D14			$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	IOR, PD
X1D15			$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	IOR, PD
X1D16	$X_0L3^1_{in}$		$4D^0$	$8B^2$	$16A^{10}$		I/O	IOL, PD
X1D17	$X_0L3^0_{in}$		$4D^1$	$8B^3$	$16A^{11}$		I/O	IOL, PD
X1D18	$X_0L3^0_{out}$		$4D^2$	$8B^4$	$16A^{12}$		I/O	IOL, PD
X1D19	$X_0L3^1_{out}$		$4D^3$	$8B^5$	$16A^{13}$		I/O	IOL, PD
X1D20			$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X1D21			$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X1D22	$X_0L3^4_{out}$	$1G^0$					I/O	IOL, PD
X1D23		$1H^0$					I/O	IOL, PD
X1D24		$1I^0$					I/O	IOR, PD
X1D25		$1J^0$					I/O	IOR, PD
X1D26			$4E^0$	$8C^0$	$16B^0$		I/O	IOT, PD
X1D27			$4E^1$	$8C^1$	$16B^1$		I/O	IOT, PD
X1D28			$4F^0$	$8C^2$	$16B^2$		I/O	IOT, PD
X1D29			$4F^1$	$8C^3$	$16B^3$		I/O	IOT, PD
X1D30			$4F^2$	$8C^4$	$16B^4$		I/O	IOT, PD
X1D31			$4F^3$	$8C^5$	$16B^5$		I/O	IOT, PD
X1D32			$4E^2$	$8C^6$	$16B^6$		I/O	IOT, PD
X1D33			$4E^3$	$8C^7$	$16B^7$		I/O	IOT, PD
X1D34	$X_0L0^2_{out}$	$1K^0$					I/O	IOL, PD
X1D35	$X_0L0^3_{out}$	$1L^0$					I/O	IOL, PD
X1D36	$X_0L0^4_{out}$	$1M^0$		$8D^0$	$16B^8$		I/O	IOL, PD
X1D37	$X_0L3^4_{in}$	$1N^0$		$8D^1$	$16B^9$		I/O	IOL, PD
X1D38	$X_0L3^3_{in}$	$1O^0$		$8D^2$	$16B^{10}$		I/O	IOL, PD
X1D39	$X_0L3^2_{in}$	$1P^0$		$8D^3$	$16B^{11}$		I/O	IOL, PD

(continued)

Signal	Function	Type	Properties
X1D40	$8D^4$ $16B^{12}$	I/O	IOT, PD
X1D41	$8D^5$ $16B^{13}$	I/O	IOT, PD
X1D42	$8D^6$ $16B^{14}$	I/O	IOT, PD
X1D43	$8D^7$ $16B^{15}$	I/O	IOT, PD

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

5 Example Application Diagram

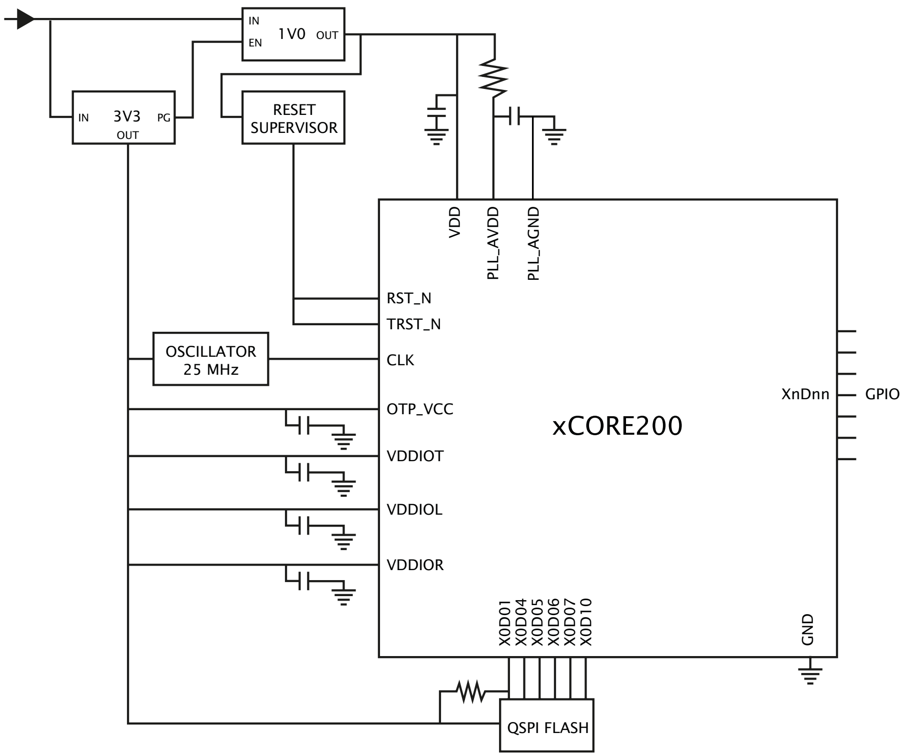


Figure 2: Simplified Reference Schematic

- see Section 11 for details on the power supplies and PCB design

6 Product Overview

The XL216-512-TQ128 is a powerful device that consists of two xCORE Tiles, each comprising a flexible logical processing cores with tightly integrated I/O and on-chip memory.

6.1 Logical cores

Each 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**
10	1000 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 XL216-512-TQ128, 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.

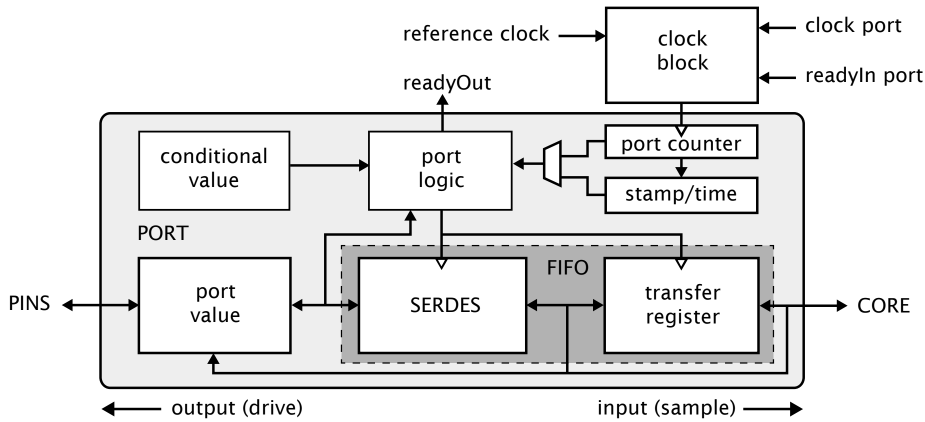


Figure 4: Port block diagram

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.

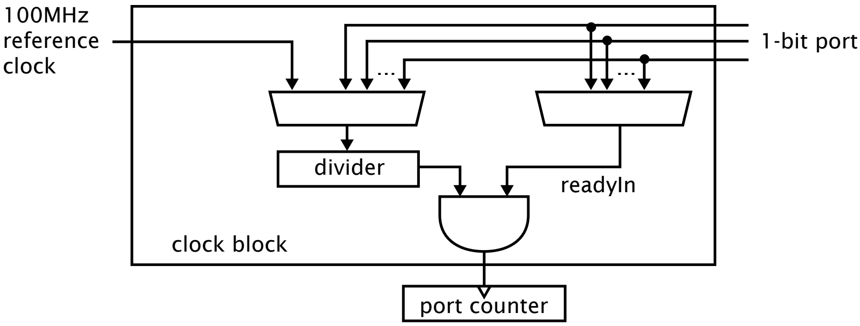


Figure 5: Clock block diagram

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

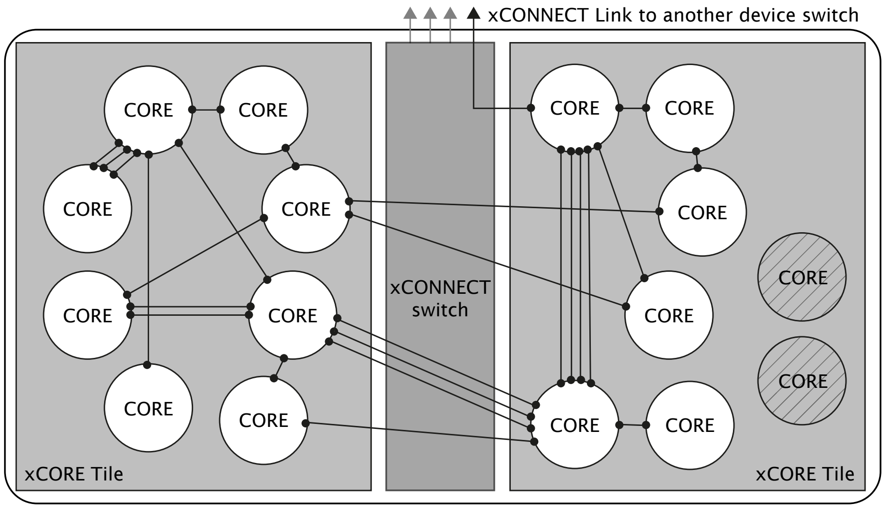


Figure 6: Switch, links and channel ends

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-L 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 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. 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 xCORE Tile boot procedure is illustrated in Figure 8. If bit 5 of the security register (*see* §9.1) is set, the device boots from OTP. To get a high value, a 3K3 pull-up resistor should be strapped onto the pin. To assure a low value, a pull-down resistor is required if other external devices are connected to this port.

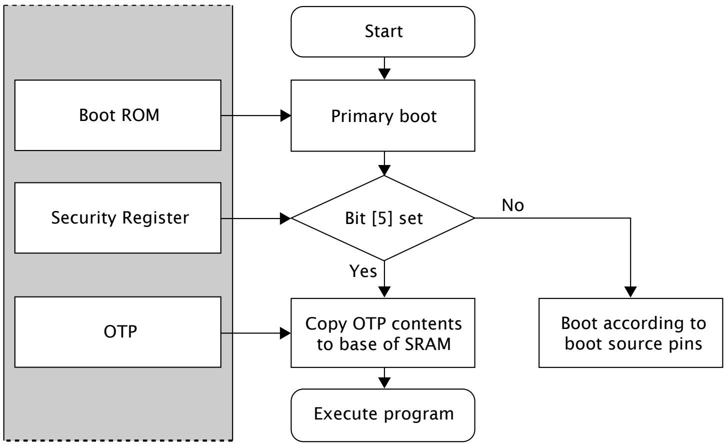


Figure 8: Boot procedure

Figure 9: Boot source pins

X0D06	X0D05	X0D04	Tile 0 boot	Tile 1 boot	Enabled links
0	0	0	QSPI master	Channel end 0	None
0	0	1	SPI master	Channel end 0	None
0	1	0	SPI slave	Channel end 0	None
0	1	1	SPI slave	SPI slave	None
1	0	0	Channel end 0	Channel end 0	XL0 (2w)

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 Boot from QSPI master

If set to boot from QSPI master, the processor enables the six pins specified in Figure 10, and drives the SPI clock at 50 MHz (assuming a 400 MHz core clock). A READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

Figure 10: QSPI pins

Pin	Signal	Description
X0D01	SS	Slave Select
X0D04..X0D07	SPIO	Data
X0D10	SCLK	Clock

The xCORE Tile expects each byte to be transferred with the *least-significant nibble first*. Programmers who write bytes into an QSPI interface using the most significant nibble first may have to reverse the nibbles in each byte of the image stored in the QSPI device.

The pins used for QSPI boot are hardcoded in the boot ROM and cannot be changed. If required, an QSPI boot program can be burned into OTP that uses different pins.

8.2 Boot from SPI master

If set to boot from SPI master, the processor enables the four pins specified in Figure 11, and drives the SPI clock at 2.5 MHz (assuming a 400 MHz core clock). A READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

Figure 11: SPI master pins

Pin	Signal	Description
X0D00	MISO	Master In Slave Out (Data)
X0D01	SS	Slave Select
X0D10	SCLK	Clock
X0D11	MOSI	Master Out Slave In (Data)

The xCORE Tile expects each byte to be transferred with the *least-significant bit first*. Programmers who write bytes into an SPI interface using the most significant bit first may have to reverse the bits in each byte of the image stored in the SPI device.

If a large boot image is to be read in, it is faster to first load a small boot-loader that reads the large image using a faster SPI clock, for example 50 MHz or as fast as the flash device supports.

The pins used for SPI boot are hardcoded in the boot ROM and cannot be changed. If required, an SPI boot program can be burned into OTP that uses different pins.

8.3 Boot from SPI slave

If set to boot from SPI slave, the processor enables the three pins specified in Figure 12 and expects a boot image to be clocked in. The supported clock polarity and phase are 0/0 and 1/1.

Pin	Signal	Description
X0D00	SS	Slave Select
X0D10	SCLK	Clock
X0D11	MOSI	Master Out Slave In (Data)

Figure 12: SPI slave pins

The xCORE Tile expects each byte to be transferred with the *least-significant bit first*. The pins used for SPI boot are hardcoded in the boot ROM and cannot be changed. If required, an SPI boot program can be burned into OTP that uses different pins.

8.4 Boot from xConnect Link

If set to boot from an xConnect Link, the processor enables its link(s) around 2 us after the boot process starts. Enabling the Link switches off the pull-down resistors on the link, drives all the TX wires low (the initial state for the Link), and monitors the RX pins for boot-traffic; they must be low at this stage. If the internal pull-down is too weak to drain any residual charge, external pull-downs of 10K may be required on those pins.

The boot-rom on the core will then:

1. Allocate channel-end 0.
2. Input a word on channel-end 0. It will use this word as a channel to acknowledge the boot. Provide the null-channel-end 0x0000FF02 if no acknowledgment is required.
3. Input the boot image specified above, including the CRC.
4. Input an END control token.
5. Output an END control token to the channel-end received in step 2.
6. Free channel-end 0.
7. Jump to the loaded code.

8.5 Boot from OTP

If an xCORE tile is set to use secure boot (see Figure 8), the boot image is read from address 0 of the OTP memory in the tile's security module.

This feature can be used to implement a secure bootloader which loads an encrypted image from external flash, decrypts and CRC checks it with the processor, and discontinues the boot process if the decryption or CRC check fails. XMOS provides a default secure bootloader that can be written to the OTP along with secret decryption keys.

Each tile has its own individual OTP memory, and hence some tiles can be booted from OTP while others are booted from SPI or the channel interface. This enables systems to be partially programmed, dedicating one or more tiles to perform a particular function, leaving the other tiles user-programmable.

8.6 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 13 provide a strong level of protection and are sufficient for providing strong IP security.

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 13: Security register features

9 Memory

9.1 OTP

Each 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

Each 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 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 JTAG

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

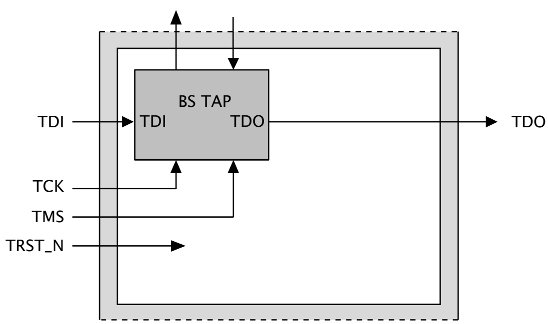


Figure 14: JTAG chain structure

The JTAG chain structure is illustrated in Figure 14. 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 TRST_N pin must be asserted low during and after power up for 100 ns. If JTAG is not required, the TRST_N pin can be tied to ground to hold the JTAG module in reset.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 15.

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			

Figure 15: IDCODE return value

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 16. The OTP User ID field is read from bits [22:31] of the security register on xCORE Tile 0, *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	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				2				8				0				0				0			

Figure 16: USERCODE return value

11 Board Integration

The device has the following power supply pins:

- VDD pins for the xCORE Tile
- VDDIO pins for the I/O lines. Separate I/O supplies are provided for the left, top, 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
- PLL_AVDD pins for the PLL
- OTP_VCC pins for the OTP

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.

11.1 Land patterns and solder stencils

The package is a 128 pin Thin Quad Flat Package (TQFP) with exposed ground paddle/heat slug on a 0.4mm 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 13 specify the dimensions and tolerances.

11.2 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.

11.3 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.

12 DC and Switching Characteristics

12.1 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	
VDDIOT 3v3	I/O supply voltage	3.135	3.30	3.465	V	
VDDIOT 2v5	I/O supply voltage	2.375	2.50	2.625	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
OTP_VCC	OTP supply voltage	3.135	3.30	3.465	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Ta	Ambient operating temperature (Commercial)	0		70	°C	
	Ambient operating temperature (Industrial)	-40		85	°C	
Tj	Junction temperature			125	°C	
Tstg	Storage temperature	-65		150	°C	

Figure 17: Operating conditions

12.2 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	

Figure 18: DC characteristics

A All pins except power supply pins.

B Pins X1D40, X1D41, X1D42, X1D43, X1D26, and X1D27 are nominal 8 mA drivers, the remainder of the general-purpose I/Os are 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.

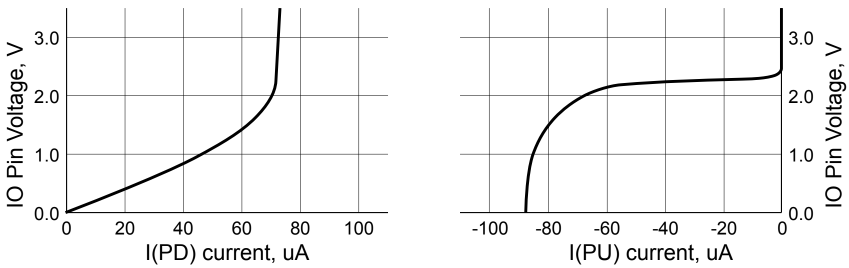


Figure 19: Typical internal pull-down and pull-up currents

12.3 ESD Stress Voltage

Figure 20: 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	

12.4 Reset Timing

Figure 21: 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.

12.5 Power Consumption

Figure 22: xCORE Tile currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
I(DDCQ)	Quiescent VDD current		45		mA	A, B, C
PD	Tile power dissipation		325		µW/MIPS	A, D, E, F
IDD	Active VDD current		570	700	mA	A, G
I(ADDPLL)	PLL_AVDD current		5	7	mA	H

A Use for budgetary purposes only.
B Assumes typical tile and I/O voltages with no switching activity.
C Includes PLL current.
D Assumes typical tile and I/O voltages with nominal switching activity.
E Assumes 1 MHz = 1 MIPS.
F PD(TYP) value is the usage power consumption under typical operating conditions.
G Measurement conditions: VDD = 1.0 V, VDDIO = 3.3 V, 25 °C, 500 MHz, average device resource usage.
H PLL_AVDD = 1.0 V

The tile power consumption of the device is highly application dependent and should be used for budgetary purposes only.