PRELIMINARY

XL212-512-TQ128 Datasheet

-XMOS®-

PRELIMINARY

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/.

PRELIMINARY

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.
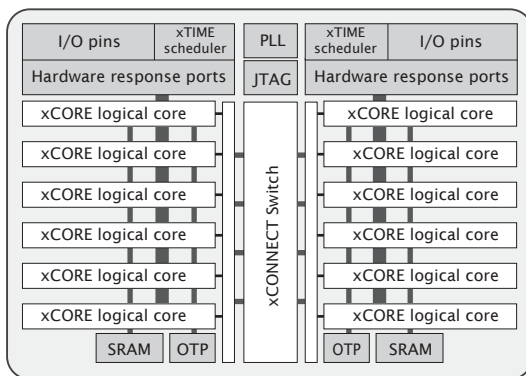
Figure 1:
XL212-512-
TQ128 block
diagram

Key features of the XL212-512-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

PRELIMINARY

▶ **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 XL212-512-TQ128 Features

▶ **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
- 12 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 $1/6$ 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
- 32 channel ends 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**
- -40 °C to 105 °C
- Commercial qualification: 0 °C to 70 °C
- Industrial qualification: -40 °C to 85 °C

▶ **Speed Grade**
- 20: 1000 MIPS

▶ **Power Consumption**
- Standby Mode
  — TBC mA

▶ **128-pin TQFP package 0.4 mm pitch**

3 Pin Configuration

Top pins (left to right): TDI 128, TCK 127, TMS 126, VDD 125, CLK 124, RST_N 123, TRST_N 122, X1D11 121, X1D10 120, VDD 119, X1D33 118, X1D32 117, X1D31 116, X1D30 115, X1D29 114, X1D28 113, X1D27 112, X1D26 111, VDDIOT 110, VDDIOT 109, X1D43 108, X1D42 107, X1D41 106, X1D40 105, OTP_VCC 104, PLL_AGND 103, PLL_AVDD 102, VDD 101, VDD 100, X0D31 99, X0D30 98, X0D29 97

Left pins (top to bottom):
Pin	Signal
1	TDO
2	X0D36 1M
3	X0D37 1N
4	VDDIOL
5	X0D38 1O
6	X0D39 1P
7	X0D40 8D
8	X0D41 8D
9	VDD
10	X0D42 8D
11	X0D43 8D
12	X1D34 1K
13	VDDIOL
14	X1D35 1L
15	X1D36 1M
16	VDD
17	VDD
18	X1D37 1N
19	VDDIOL
20	X1D38 1O
21	X1D39 1P
22	X1D16 4D
23	X1D17 4D
24	VDD
25	X1D18 4D
26	X1D19 4D
27	X0D01 1B
28	X0D10 1C
29	VDDIOL
30	X1D22 1G
31	X1D23 1H
32	X0D00 1A

Right pins (top to bottom):
Pin	Signal
96	X0D33 4E
95	X0D28 4F
94	X0D32 4E
93	X0D27 4E
92	X0D26 4E
91	VDDIOR
90	X0D35 1L
89	X0D34 1K
88	X0D25 1J
87	X0D24 1I
86	VDD
85	X1D01 1B
84	X1D00 1A
83	X1D09 4A
82	VDDIOR
81	X1D08 4A
80	VDD
79	VDD
78	X1D07 4B
77	VDDIOR
76	X1D06 4B
75	X1D05 4B
74	X1D04 4B
73	X1D03 4A
72	VDD
71	X1D02 4A
70	X0D19 4D
69	X0D18 4D
68	X0D17 4D
67	X0D16 4D
66	VDDIOR
65	NC

Bottom pins (left to right): X0D11 33 1D, X0D04 34 4B, X0D05 35 4B, X0D06 36 4B, X0D07 37 4B, VDD 38, X0D02 39 4A, X0D03 40 4A, X0D08 41 4A, VDD 42, X0D09 43 4A, X1D12 44 1E, X1D13 45 1F, VDDIOL 46, VDD 47, VDDIOR 48, X1D14 49 4C, X1D15 50 4C, X1D20 51 4C, X1D21 52 4C, X1D24 53 1I, X1D25 54 1J, VDD 55, X0D14 56 4C, X0D15 57 4C, X0D20 58 4C, VDD 59, X0D21 60 4C, X0D12 61 1E, X0D13 62 1F, X0D22 63 1G, X0D23 64 1H

SUBJECT TO CHANGE

PRELIMINARY

4 Signal Description

This section lists the signals and I/O pins available on the XL212-512-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 a weak pull-down or pull-up resistor. On GPIO pins this resistor can be enabled.

▶ 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	$XL3^2_{out}$	$1B^0$				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

(continued)

Signal	Function					Type	Properties
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	$XL3_{out}^3$	$1C^0$				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	$XL4_{in}^4$	$4D^0$	$8B^2$	$16A^{10}$		I/O	IOR, PD
X0D17	$XL4_{in}^3$	$4D^1$	$8B^3$	$16A^{11}$		I/O	IOR, PD
X0D18	$XL4_{in}^2$	$4D^2$	$8B^4$	$16A^{12}$		I/O	IOR, PD
X0D19	$XL4_{in}^1$	$4D^3$	$8B^5$	$16A^{13}$		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	$XL7_{in}^0$	$1I^0$				I/O	IOR, PD
X0D25	$XL7_{out}^0$	$1J^0$				I/O	IOR, PD
X0D26	$XL7_{out}^3$	$4E^0$	$8C^0$	$16B^0$		I/O	IOR, PD
X0D27	$XL7_{out}^4$	$4E^1$	$8C^1$	$16B^1$		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	$XL7_{out}^1$	$1K^0$				I/O	IOR, PD
X0D35	$XL7_{out}^2$	$1L^0$				I/O	IOR, PD
X0D36		$1M^0$	$8D^0$	$16B^8$		I/O	IOL, PD
X0D37	$XL0_{in}^4$	$1N^0$	$8D^1$	$16B^9$		I/O	IOL, PD
X0D38	$XL0_{in}^3$	$1O^0$	$8D^2$	$16B^{10}$		I/O	IOL, PD
X0D39	$XL0_{in}^2$	$1P^0$	$8D^3$	$16B^{11}$		I/O	IOL, PD
X0D40	$XL0_{in}^1$		$8D^4$	$16B^{12}$		I/O	IOL, PD
X0D41	$XL0_{in}^0$		$8D^5$	$16B^{13}$		I/O	IOL, PD
X0D42	$XL0_{out}^0$		$8D^6$	$16B^{14}$		I/O	IOL, PD
X0D43	$XL0_{out}^1$		$8D^7$	$16B^{15}$		I/O	IOL, PD
X1D00	$XL7_{in}^2$	$1A^0$				I/O	IOR, PD
X1D01	$XL7_{in}^1$	$1B^0$				I/O	IOR, PD
X1D02	$XL4_{in}^0$	$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	IOR, PD

(continued)

Signal	Function				Type	Properties
X1D03	$XL4_{out}^0$	$4A^1$	$8A^1$	$16A^1$ $32A^{21}$	I/O	IOR, PD
X1D04	$XL4_{out}^1$	$4B^0$	$8A^2$	$16A^2$ $32A^{22}$	I/O	IOR, PD
X1D05	$XL4_{out}^2$	$4B^1$	$8A^3$	$16A^3$ $32A^{23}$	I/O	IOR, PD
X1D06	$XL4_{out}^3$	$4B^2$	$8A^4$	$16A^4$ $32A^{24}$	I/O	IOR, PD
X1D07	$XL4_{out}^4$	$4B^3$	$8A^5$	$16A^5$ $32A^{25}$	I/O	IOR, PD
X1D08	$XL7_{in}^4$	$4A^2$	$8A^6$	$16A^6$ $32A^{26}$	I/O	IOR, PD
X1D09	$XL7_{in}^3$	$4A^3$	$8A^7$	$16A^7$ $32A^{27}$	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	$XL3_{in}^1$	$4D^0$	$8B^2$	$16A^{10}$	I/O	IOL, PD
X1D17	$XL3_{in}^0$	$4D^1$	$8B^3$	$16A^{11}$	I/O	IOL, PD
X1D18	$XL3_{out}^0$	$4D^2$	$8B^4$	$16A^{12}$	I/O	IOL, PD
X1D19	$XL3_{out}^1$	$4D^3$	$8B^5$	$16A^{13}$	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	$XL3_{out}^4$	$1G^0$			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	$XL0_{out}^2$	$1K^0$			I/O	IOL, PD
X1D35	$XL0_{out}^3$	$1L^0$			I/O	IOL, PD
X1D36	$XL0_{out}^4$	$1M^0$	$8D^0$	$16B^8$	I/O	IOL, PD
X1D37	$XL3_{in}^4$	$1N^0$	$8D^1$	$16B^9$	I/O	IOL, PD
X1D38	$XL3_{in}^3$	$1O^0$	$8D^2$	$16B^{10}$	I/O	IOL, PD
X1D39	$XL3_{in}^2$	$1P^0$	$8D^3$	$16B^{11}$	I/O	IOL, PD
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

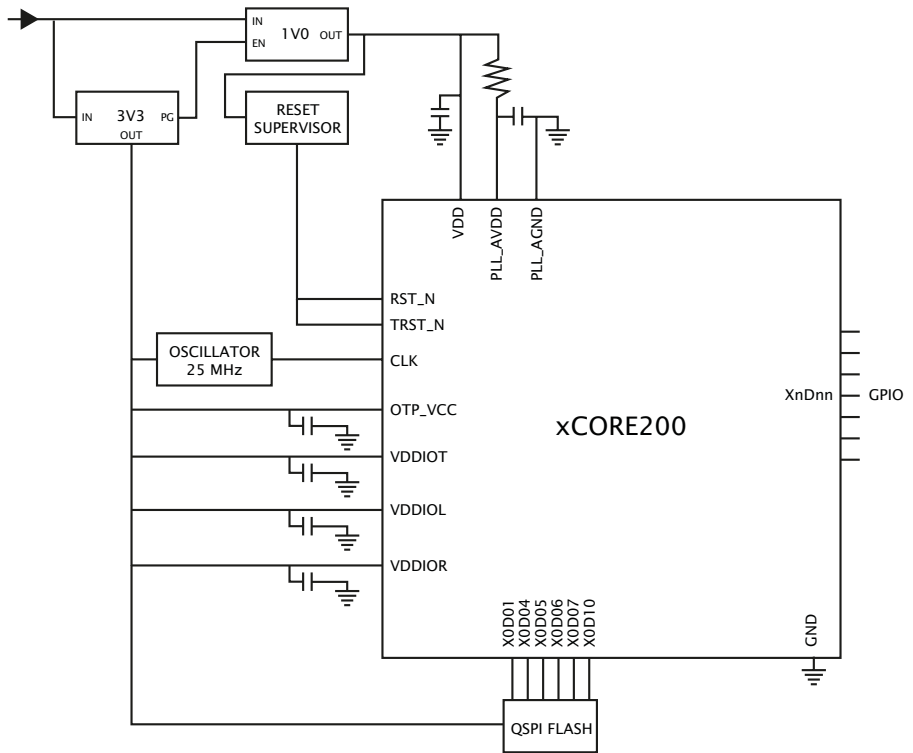| System pins (1)			
Signal	**Function**	**Type**	**Properties**
CLK	PLL reference clock	Input	IOL, PD, ST

5 Example Application Diagram

Figure 2:
Simplified
Reference
Schematic

PRELIMINARY

6 Product Overview

The XL212-512-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 6 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		
10	1000 MIPS	500 MHz	100	100	100	100	100	83		

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 XL212-512-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.
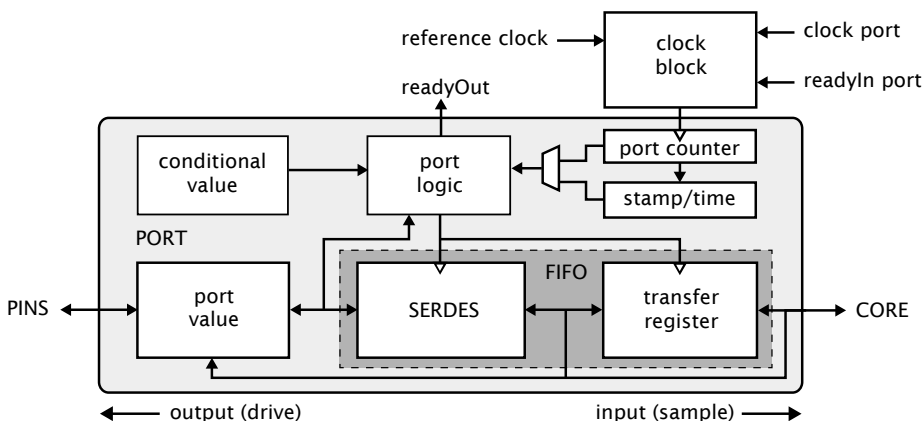

Figure 4:
Port block
diagram

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.
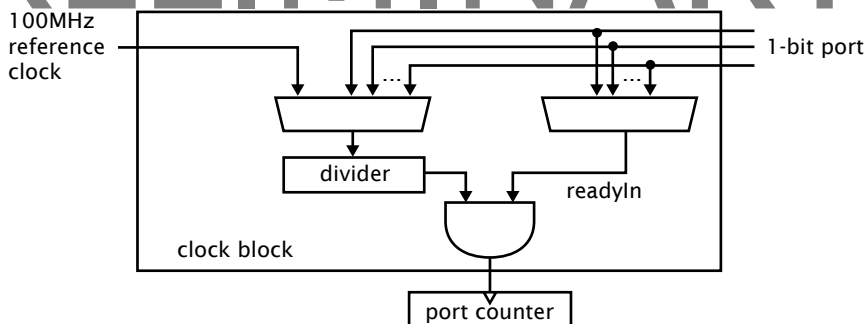
PRELIMINARY

100MHz
reference
clock

1-bit port

divider

readyIn

clock block

port counter

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
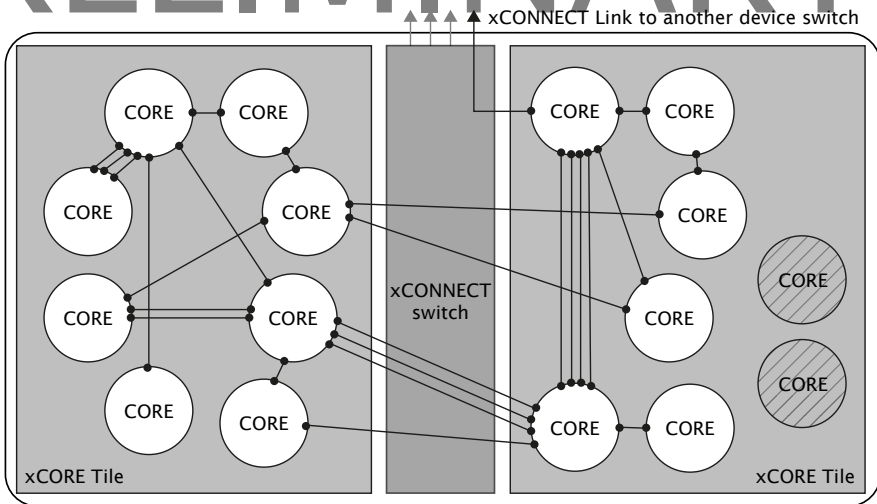
PRELIMINARY

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

XMOS

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.
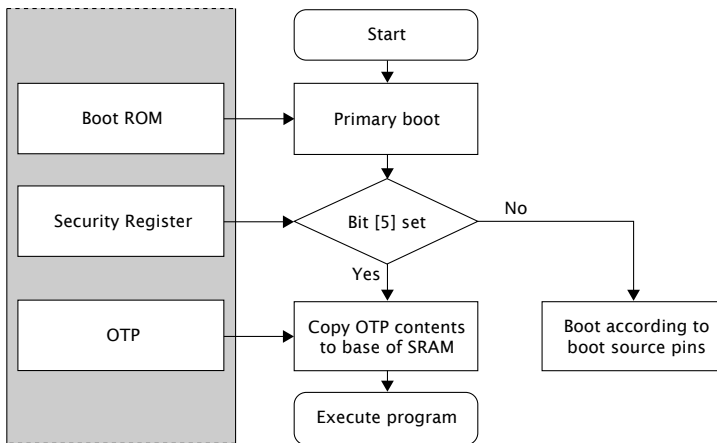

Figure 8:
Boot
procedure

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)
1	0	1	Channel end 0	Channel end 0	XL4-XL7 (5w)
1	1	0	Channel end 0	Channel end 0	XL1, XL2, XL5, and XL6 (5w)
1	1	1	Channel end 0	Channel end 0	XL0-XL3 (5w)

Figure 9:
Boot source
pins

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.

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

Figure 10: QSPI pins

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.

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

Figure 11: SPI master pins

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.

Figure 12:
SPI slave pins

Pin	Signal	Description
X0D00	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.* 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.
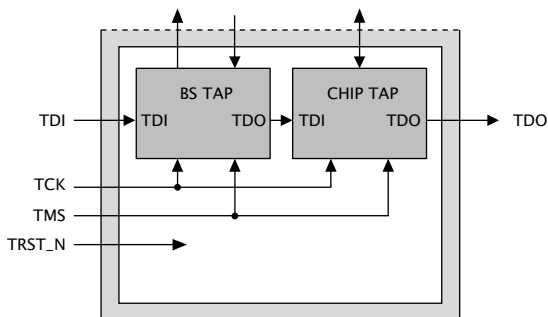

Figure 14:
JTAG chain
structure

The JTAG chain structure is illustrated in Figure 14. Directly after reset, two TAP controllers are present in the JTAG chain for each xCORE Tile: the boundary scan TAP and the chip TAP. The boundary scan TAP is a standard 1149.1 compliant TAP

that can be used for boundary scan of the I/O pins. The chip TAP provides access into the xCORE Tile, switch and OTP 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.

Figure 15:
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 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 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).

Figure 16:
USERCODE
return value

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	0	0	2	8	0	0	0

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 PLLVDD 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, 4x100nF 0402 low inductance MLCCs per supply rail). 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 land pattern recommendations in this document are based on a RoHS compliant process and derived, where possible, from the nominal *Generic Requirements for Surface Mount Design and Land Pattern Standards* IPC-7351B specifications. This standard aims to achieve desired targets of heel, toe and side fillets for solder-joints.

Solder paste and ground via recommendations are based on our engineering and development kit board production. They have been found to work and optimized as appropriate to achieve a high yield. The size, type and number of vias used in the center pad affects how much solder wicks down the vias during reflow. This in turn, along with solder paster coverage, affects the final assembled package height. These factors should be taken into account during design and manufacturing of the PCB.

The following land patterns and solder paste contains recommendations. Final land pattern and solder paste decisions are the responsibility of the customer. These should be tuned during manufacture to suit the manufacturing process.

The package is a 128 pin Thin Quad Flat Pack package with exposed heat slug on a 0.4mm pitch. An example land pattern is shown in Figure 17.

Pad widths and spacings are such that solder mask can still be applied between the pads using standard design rules. This is recommended to reduce solder shorts.

The center pad solder paste level needs to be controlled so the device sits the correct height from the circuit board. For the 128 pin TQFP package, a 3x3 array of squares for solder paste is recommended as shown in Figure 18. This gives a paste level of 56%.

PRELIMINARY
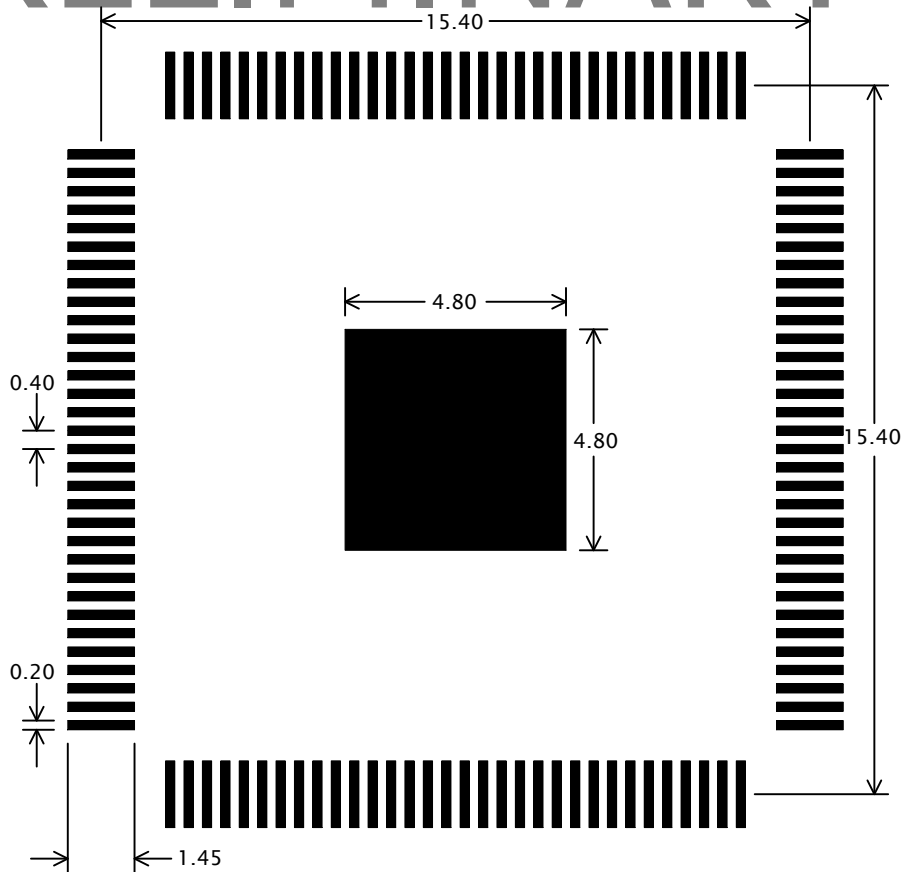
Figure 17:
Example land
pattern

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. A 3 x 3 grid of vias, with a 0.6mm diameter annular ring and a 0.3mm drill, equally spaced across the heat slug, would be suitable.

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.

PRELIMINARY
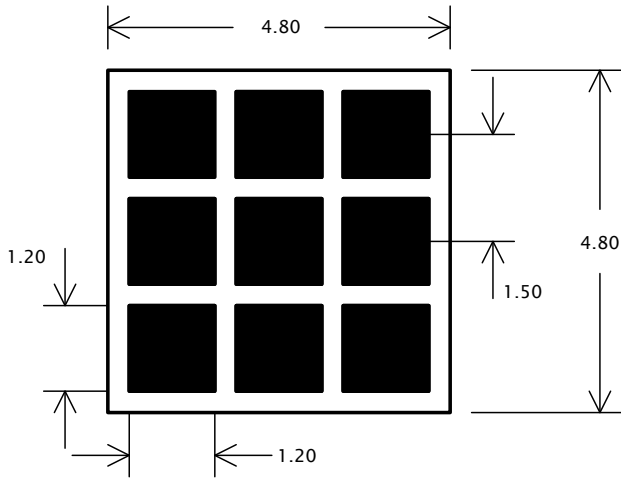
All XMOS devices are Moisture Sensitivity Level (MSL) 3 - devices have a shelf life of 168 hours between removal from the packaging and reflow, provided they are stored below 30C and 60% RH. If devices have exceeded these values or an included moisture indicator card shows excessive levels of moisture, 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	2.25	3.30	3.60	V	
VDDIOR	I/O supply voltage	2.25	3.30	3.60	V	
VDDIOT	I/O supply voltage	2.25	3.30	3.60	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
OTP_VCC	OTP supply voltage	3.00	3.30	3.60	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Ta	Ambient operating temperature (Commercial)	0		70	℃	
	Ambient operating temperature (Industrial)	-40		85	℃	
	Ambient operating temperature ()	-40		105	℃	
Tj	Junction temperature			125	℃	
Tstg	Storage temperature	-65		150	℃	

Figure 19:
Operating conditions

12.2 DC Characteristics

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	2.00			V	B, C
V(OL)	Output low voltage			0.60	V	B, C
R(PU)	Pull-up resistance		35K		Ω	D
R(PD)	Pull-down resistance		35K		Ω	D

Figure 20:
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.

12.3 ESD Stress Voltage

Figure 21:
ESD stress voltage

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
HBM	Human body model	-2.00		2.00	KV	
MM	Machine model	-200		200	V	

12.4 Reset Timing

Symbol	Parameters	MIN	TYP	MAX	UNITS	Notes
T(RST)	Reset pulse width	5			us	
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

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

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, 400 MHz, average device resource usage.
H Measurement conditions: VDD = 1.0 V, VDDIO = 3.3 V, 25 °C, 500 MHz, average device resource usage.
I PLL_AVDD = 1.0 V

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

More detailed power analysis can be found in the XS1-L Power Consumption document,

12.6 Clock

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f	Frequency	9	25	25	MHz	
SR	Slew rate	0.10			V/ns	
TJ(LT)	Long term jitter (pk-pk)			2	%	A
f(MAX)	Processor clock frequency ()			400	MHz	B
	Processor clock frequency			500	MHz	B

A Percentage of CLK period.
B Assumes typical tile and I/O voltages with nominal activity.

Further details can be found in the XS1-L Clock Frequency Control document,

PRELIMINARY

12.7   xCORE Tile I/O AC Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
T(XOVALID)	Input data valid window	8			ns	
T(XOINVALID)	Output data invalid window	9			ns	
T(XIFMAX)	Rate at which data can be sampled with respect to an external clock			60	MHz	

The input valid window parameter relates to the capability of the device to capture data input to the chip with respect to an external clock source. It is calculated as the sum of the input setup time and input hold time with respect to the external clock as measured at the pins. The output invalid window specifies the time for which an output is invalid with respect to the external clock. Note that these parameters are specified as a window rather than absolute numbers since the device provides functionality to delay the incoming clock with respect to the incoming data.

Information on interfacing to high-speed synchronous interfaces can be found in the XS1 Port I/O Timing document, X5821.

12.8   xConnect Link Performance

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
B(2blinkP)	2b link bandwidth (packetized)			87	MBit/s	A, B
B(5blinkP)	5b link bandwidth (packetized)			217	MBit/s	A, B
B(2blinkS)	2b link bandwidth (streaming)			100	MBit/s	B
B(5blinkS)	5b link bandwidth (streaming)			250	MBit/s	B

A   Assumes 32-byte packet in 3-byte header mode. Actual performance depends on size of the header and payload.

B   7.5 ns symbol time.

The asynchronous nature of links means that the relative phasing of CLK clocks is not important in a multi-clock system, providing each meets the required stability criteria.

12.9   JTAG Timing

| Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			18	MHz	
f(TCK_B)	TCK frequency (boundary scan)			10	MHz	
T(SETUP)	TDO to TCK setup time	5			ns	A
T(HOLD)	TDO to TCK hold time	5			ns	A
T(DELAY)	TCK to output delay			15	ns	B

A   Timing applies to TMS and TDI inputs.

B   Timing applies to TDO output from negative edge of TCK.

All JTAG operations are synchronous to TCK apart from the global asynchronous reset TRST_N.

13 Package Information
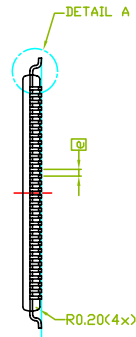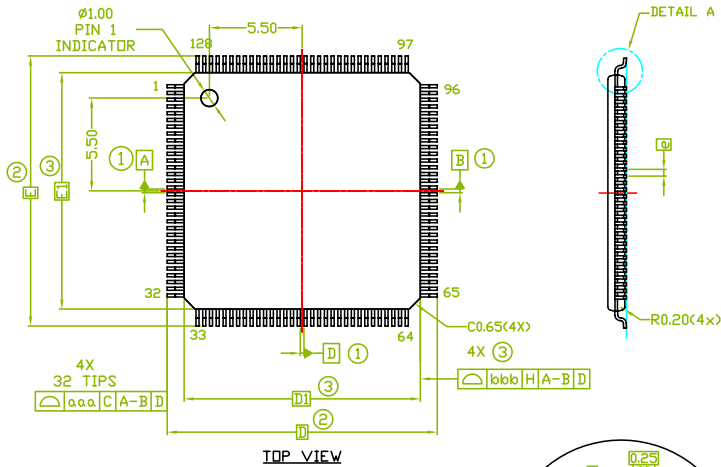
SYMBOL	Min.	Nom.	Max.
A	–	–	1.20
A1	0.05	–	0.15
A2	0.95	1.00	1.05
b	0.13	0.18	0.23
b1	0.13	0.16	0.19
D	16.00 BSC		
D1	14.00 BSC		
e	0.40 BSC		
E	16.00 BSC		
E1	14.00 BSC		
θ	0°	3.5°	7°
θ1	0°	–	–
θ2	11°	12°	13°
θ3	11°	12°	13°
c	0.09	–	0.20
c1	0.09	–	0.16
L	0.45	0.60	0.75
L1	1.00 REF		
R1	0.08	–	–
R2	0.08	–	0.20
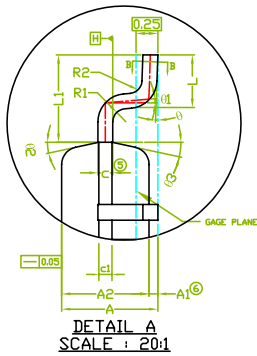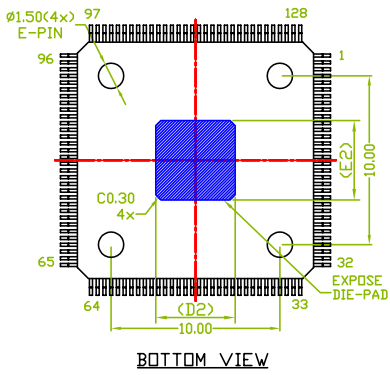
REF	TOLERANCES OF FORM AND POSITION
aaa	0.20
bbb	0.20
ccc	0.08
ddd	0.07
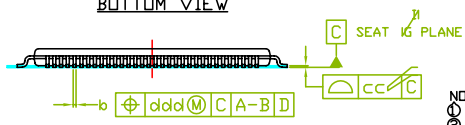
LF Ref#	Symbol	Min	Nom	Max
L-17-09011	D2	4.60	4.70	4.80
	E2	4.60	4.70	4.80

NOTE :
① DATUM A-B AND D TO DETERMINE AT DATUM PLANE H.
② TO BE DETERMINED AT SEATING DATUM PLAN C.
③ DIMENSION D1 AND E1 DO NOT INCLUDE MOLD PROTRUSIONS. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. S1 AND E1 ARE MAXIMUM PLASTIC BODY SIZE DIMENSION INCLUDING MOLD MISMATCH.
④ DIMENSION b DOSE NOT INCLUDE DAMBAR PROTRUSION. ALLOWABLE DAMBAR PROTRUSION SHALL NOT CAUSE THE LEAD WIDTH TO EXCEED THE MAXIMUM b DIMENSION BY MORE THAN 0.08 mm. DAMBAR CANNOT BE LOCATED ON THE LOWER RADIUS OR THE FOOT. MINIMUM SPACE BETWEEN PROTRUSION AND ADJACENT LEAD IS 0.07 mm FOR 0.4mm AND 0.5 mm PITCH PACKAGE.
⑤ THESE DIMENSIONS APPLY TO THE FLAT SECTION OF THE BETWEEN 0.10 mm AND 0.25 mm FROM THE LEAD TIP.
⑥ A1 IS THE DEFINED AS THE DISTANCE FROM THE SEATING PLAN TO THE LOWEST POINT ON THE PACKAGE BODY.
⑦ PACKAGE LEAD COUNT IS NON-JEDEC STANDARD.

PRELIMINARY

13.1 Part Marking

FXCCRNTM

MCYYWWXX

LLLLLL.LL

F - Product family
X - Reserved
CC - Number of logical cores
R - RAM (in log-2)
N - Flash size
T - Temperature grade
M - MHz grade

MC - Manufacturer
YYWW - Date
XX - Reserved

Wafer lot code

Figure 28:
Part marking
scheme

14 Ordering Information

Figure 29:
Orderable
part numbers

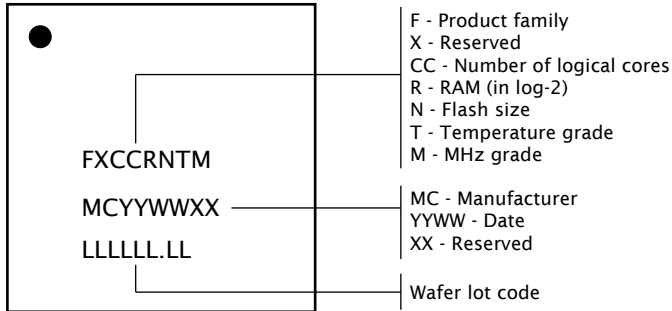| Product Code	Marking	Qualification	Speed Grade
XL212-512-TQ128-C20	L20880C5	Commercial	1000 MIPS
XL212-512-TQ128-I20	L20880I5	Industrial	1000 MIPS

Appendices

A   Configuration of the XL212-512-TQ128

The device is configured through banks of registers, as shown in Figure 30.
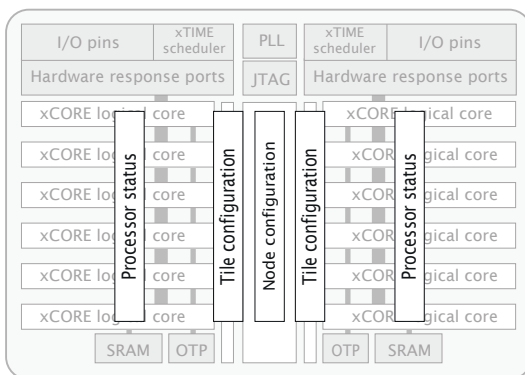
Figure 30:
Registers

The following communication sequences specify how to access those registers. Any messages transmitted contain the most significant 24 bits of the channel-end to which a response is to be sent. This comprises the node-identifier and the channel number within the node. if no response is required on a write operation, supply 24-bits with the last 8-bits set, which suppresses the reply message. Any multi-byte data is sent most significant byte first.

A.1   Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0C. Alternatively, the functions `getps(reg)` and `setps(reg,value)` can be used from XC.

A.2   Accessing an xCORE Tile configuration register

xCORE Tile configuration registers can be accessed through the interconnect using the functions `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tile ↪ ref, ...)`, where `tileref` is the name of the xCORE Tile, e.g. `tile[1]`. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the xCORE tile configuration registers. The destination of the channel-end should be set to `0xnnnnnC20C` where `nnnnnn` is the tile-identifier.

A write message comprises the following:

control-token	24-bit response	16-bit	32-bit	control-token
192	channel-end identifier	register number	data	1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token	24-bit response	16-bit	control-token
193	channel-end identifier	register number	1

The response to the read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

A.3   Accessing node configuration

Node configuration registers can be accessed through the interconnect using the functions `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ↪ ...)`, where `device` is the name of the node. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the node configuration registers. The destination of the channel-end should be set to `0xnnnnC30C` where `nnnn` is the node-identifier.

A write message comprises the following:

control-token	24-bit response	16-bit	32-bit	control-token
192	channel-end identifier	register number	data	1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token	24-bit response	16-bit	control-token
193	channel-end identifier	register number	1

The response to a read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

B Processor Status Configuration

The processor status control registers can be accessed directly by the processor using processor status reads and writes (use `getps(reg)` and `setps(reg,value)` for reads and writes).

Number	Perm	Description
0x00	RW	RAM base address
0x01	RW	Vector base address
0x02	RW	xCORE Tile control
0x03	RO	xCORE Tile boot status
0x05	RW	Security configuration
0x06	RW	Ring Oscillator Control
0x07	RO	Ring Oscillator Value
0x08	RO	Ring Oscillator Value
0x09	RO	Ring Oscillator Value
0x0A	RO	Ring Oscillator Value
0x0C	RO	RAM size
0x10	DRW	Debug SSR
0x11	DRW	Debug SPC
0x12	DRW	Debug SSP
0x13	DRW	DGETREG operand 1
0x14	DRW	DGETREG operand 2
0x15	DRW	Debug interrupt type
0x16	DRW	Debug interrupt data
0x18	DRW	Debug core control
0x20 .. 0x27	DRW	Debug scratch
0x30 .. 0x33	DRW	Instruction breakpoint address
0x40 .. 0x43	DRW	Instruction breakpoint control
0x50 .. 0x53	DRW	Data watchpoint address 1
0x60 .. 0x63	DRW	Data watchpoint address 2
0x70 .. 0x73	DRW	Data breakpoint control register
0x80 .. 0x83	DRW	Resources breakpoint mask
0x90 .. 0x93	DRW	Resources breakpoint value
0x9C .. 0x9F	DRW	Resources breakpoint control register

Figure 31:
Summary

B.1 RAM base address: 0x00

This register contains the base address of the RAM. It is initialized to 0x00040000.

0x00:
RAM base
address

Bits	Perm	Init	Description
31:2	RW		Most significant 16 bits of all addresses.
1:0	RO	-	Reserved

B.2 Vector base address: 0x01

Base address of event vectors in each resource. On an interrupt or event, the 16 most significant bits of the destination address are provided by this register; the least significant 16 bits come from the event vector.

0x01:
Vector base
address

Bits	Perm	Init	Description
31:18	RW		The event and interrupt vectors.
17:0	RO	-	Reserved

B.3 xCORE Tile control: 0x02

Register to control features in the xCORE tile

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:18	RW	0	RGMII TX data delay value (in PLL output cycle increments)
17:9	RW	0	RGMII TX clock divider value. TX clk rises when counter (clocked by PLL output) reaches this value and falls when counter reaches (value»1). Value programmed into this field should be actual divide value required minus 1
8	RW	0	Enable RGMII interface periph ports
7:6	RO	-	Reserved
5	RW	0	Select the dynamic mode (1) for the clock divider when the clock divider is enabled. In dynamic mode the clock divider is only activated when all active threads are paused. In static mode the clock divider is always enabled.
4	RW	0	Enable the clock divider. This divides the output of the PLL to facilitate one of the low power modes.
3:0	RO	-	Reserved

0x02:
xCORE Tile
control

B.4   xCORE Tile boot status: 0x03

This read-only register describes the boot status of the xCORE tile.

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Processor number.
15:9	RO	-	Reserved
8	RO		Overwrite BOOT_MODE.
7:6	RO	-	Reserved
5	RO		Indicates if core1 has been powered off
4	RO		Cause the ROM to not poll the OTP for correct read levels
3	RO		Boot ROM boots from RAM
2	RO		Boot ROM boots from JTAG
1:0	RO		The boot PLL mode pin value.

0x03:
xCORE Tile
boot status

B.5   Security configuration: 0x05

Copy of the security register as read from OTP.

Bits	Perm	Init	Description
31	RW		Disables write permission on this register
30:15	RO	-	Reserved
14	RW		Disable access to XCore's global debug
13	RO	-	Reserved
12	RW		lock all OTP sectors
11:8	RW		lock bit for each OTP sector
7	RW		Enable OTP reduanacy
6	RO	-	Reserved
5	RW		Override boot mode and read boot image from OTP
4	RW		Disable JTAG access to the PLL/BOOT configuration registers
3:1	RO	-	Reserved
0	RW		Disable access to XCore's JTAG debug TAP

0x05:
Security
configuration

PRELIMINARY

B.6 Ring Oscillator Control: 0x06

There are four free-running oscillators that clock four counters. The oscillators can be started and stopped using this register. The counters should only be read when the ring oscillator has been stopped for at least 10 core clock cycles (this can be achieved by inserting two nop instructions between the SETPS and GETPS). The counter values can be read using four subsequent registers. The ring oscillators are asynchronous to the xCORE tile clock and can be used as a source of random bits.

0x06:
Ring
Oscillator
Control

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Core ring oscillator enable.
0	RW	0	Peripheral ring oscillator enable.

B.7 Ring Oscillator Value: 0x07

This register contains the current count of the xCORE Tile Cell ring oscillator. This value is not reset on a system reset.

0x07:
Ring
Oscillator
Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.8 Ring Oscillator Value: 0x08

This register contains the current count of the xCORE Tile Wire ring oscillator. This value is not reset on a system reset.

0x08:
Ring
Oscillator
Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.9 Ring Oscillator Value: 0x09

This register contains the current count of the Peripheral Cell ring oscillator. This value is not reset on a system reset.

PRELIMINARY

	Bits	Perm	Init	Description
0x09: Ring Oscillator Value	31:16	RO	-	Reserved
	15:0	RO	0	Ring oscillator Counter data.

B.10 Ring Oscillator Value: 0x0A

This register contains the current count of the Peripheral Wire ring oscillator. This value is not reset on a system reset.

	Bits	Perm	Init	Description
0x0A: Ring Oscillator Value	31:16	RO	-	Reserved
	15:0	RO	0	Ring oscillator Counter data.

B.11 RAM size: 0x0C

The size of the RAM in bytes

	Bits	Perm	Init	Description
0x0C: RAM size	31:2	RO		Most significant 16 bits of all addresses.
	1:0	RO	-	Reserved

B.12 Debug SSR: 0x10

This register contains the value of the SSR register when the debugger was called.

PRELIMINARY

Bits	Perm	Init	Description
31:11	RO	-	Reserved
10	DRW		Address space indentifier
9	DRW		Determines the issue mode (DI bit) upon Kernel Entry after Exception or Interrupt.
8	RO		Determines the issue mode (DI bit).
7	DRW		When 1 the thread is in fast mode and will continually issue.
6	DRW		When 1 the thread is paused waiting for events, a lock or another resource.
5	RO	-	Reserved
4	DRW		1 when in kernel mode.
3	DRW		1 when in an interrupt handler.
2	DRW		1 when in an event enabling sequence.
1	DRW		When 1 interrupts are enabled for the thread.
0	DRW		When 1 events are enabled for the thread.

0x10:
Debug SSR

B.13   Debug SPC: 0x11

This register contains the value of the SPC register when the debugger was called.

0x11:
Debug SPC

Bits	Perm	Init	Description
31:0	DRW		Value.

B.14   Debug SSP: 0x12

This register contains the value of the SSP register when the debugger was called.

0x12:
Debug SSP

Bits	Perm	Init	Description
31:0	DRW		Value.

B.15   DGETREG operand 1: 0x13

The resource ID of the logical core whose state is to be read.

	Bits	Perm	Init	Description
0x13: DGETREG operand 1	31:8	RO	-	Reserved
	7:0	DRW		Thread number to be read

B.16 DGETREG operand 2: 0x14

Register number to be read by DGETREG

	Bits	Perm	Init	Description
0x14: DGETREG operand 2	31:5	RO	-	Reserved
	4:0	DRW		Register number to be read

B.17 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

	Bits	Perm	Init	Description
	31:18	RO	-	Reserved
	17:16	DRW		Number of the hardware breakpoint/watchpoint which caused the interrupt (always 0 for =HOST= and =DCALL=). If multiple breakpoints/watchpoints trigger at once, the lowest number is taken.
	15:8	DRW		Number of thread which caused the debug interrupt (always 0 in the case of =HOST=).
	7:3	RO	-	Reserved
0x15: Debug interrupt type	2:0	DRW	0	Indicates the cause of the debug interrupt 1: Host initiated a debug interrupt through JTAG 2: Program executed a DCALL instruction 3: Instruction breakpoint 4: Data watch point 5: Resource watch point

B.18 Debug interrupt data: 0x16

On a data watchpoint, this register contains the effective address of the memory operation that triggered the debugger. On a resource watchpoint, it countains the resource identifier.

0x16:
Debug
interrupt data

Bits	Perm	Init	Description
31:0	DRW		Value.

B.19   Debug core control: 0x18

This register enables the debugger to temporarily disable logical cores. When returning from the debug interrupts, the cores set in this register will not execute. This enables single stepping to be implemented.

0x18:
Debug core
control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		1-hot vector defining which threads are stopped when not in debug mode. Every bit which is set prevents the respective thread from running.

B.20   Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over JTAG. This is the same set of registers as the Debug Scratch registers in the xCORE tile configuration.

0x20 .. 0x27:
Debug
scratch

Bits	Perm	Init	Description
31:0	DRW		Value.

B.21   Instruction breakpoint address: 0x30 .. 0x33

This register contains the address of the instruction breakpoint. If the PC matches this address, then a debug interrupt will be taken. There are four instruction breakpoints that are controlled individually.

0x30 .. 0x33:
Instruction
breakpoint
address

Bits	Perm	Init	Description
31:0	DRW		Value.

B.22   Instruction breakpoint control: 0x40 .. 0x43

This register controls which logical cores may take an instruction breakpoint, and under which condition.

0x40 .. 0x43:
Instruction
breakpoint
control

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:2	RO	-	Reserved
1	DRW	0	When 0 break when PC == IBREAK_ADDR. When 1 = break when PC != IBREAK_ADDR.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.23   Data watchpoint address 1: 0x50 .. 0x53

This set of registers contains the first address for the four data watchpoints.

0x50 .. 0x53:
Data
watchpoint
address 1

Bits	Perm	Init	Description
31:0	DRW		Value.

B.24   Data watchpoint address 2: 0x60 .. 0x63

This set of registers contains the second address for the four data watchpoints.

0x60 .. 0x63:
Data
watchpoint
address 2

Bits	Perm	Init	Description
31:0	DRW		Value.

B.25   Data breakpoint control register: 0x70 .. 0x73

This set of registers controls each of the four data watchpoints.

0x70 .. 0x73:
Data
breakpoint
control
register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:3	RO	-	Reserved
2	DRW	0	When 1 the breakpoints will be be triggered on loads.
1	DRW	0	Determines the break condition: 0 = A AND B, 1 = A OR B.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.26   Resources breakpoint mask: 0x80 .. 0x83

This set of registers contains the mask for the four resource watchpoints.

0x80 .. 0x83:
Resources
breakpoint
mask

Bits	Perm	Init	Description
31:0	DRW		Value.

B.27   Resources breakpoint value: 0x90 .. 0x93

This set of registers contains the value for the four resource watchpoints.

0x90 .. 0x93:
Resources
breakpoint
value

Bits	Perm	Init	Description
31:0	DRW		Value.

B.28   Resources breakpoint control register: 0x9C .. 0x9F

This set of registers controls each of the four resource watchpoints.

PRELIMINARY

0x9C .. 0x9F:
Resources
breakpoint
control
register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:2	RO	-	Reserved
1	DRW	0	When 0 break when condition A is met. When 1 = break when condition B is met.
0	DRW	0	When 1 the instruction breakpoint is enabled.

PRELIMINARY

C Tile Configuration

The xCORE Tile control registers can be accessed using configuration reads and writes (use `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tileref,` ↪ `...)` for reads and writes).

Number	Perm	Description
0x00	CRO	Device identification
0x01	CRO	xCORE Tile description 1
0x02	CRO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	CRW	xCORE Tile clock divider
0x07	CRO	Security configuration
0x20 .. 0x27	CRW	Debug scratch
0x40	CRO	PC of logical core 0
0x41	CRO	PC of logical core 1
0x42	CRO	PC of logical core 2
0x43	CRO	PC of logical core 3
0x44	CRO	PC of logical core 4
0x45	CRO	PC of logical core 5
0x46	CRO	PC of logical core 6
0x47	CRO	PC of logical core 7
0x60	CRO	SR of logical core 0
0x61	CRO	SR of logical core 1
0x62	CRO	SR of logical core 2
0x63	CRO	SR of logical core 3
0x64	CRO	SR of logical core 4
0x65	CRO	SR of logical core 5
0x66	CRO	SR of logical core 6
0x67	CRO	SR of logical core 7

Figure 32:
Summary

C.1 Device identification: 0x00

This register identifies the xCORE Tile

Bits	Perm	Init	Description
31:24	CRO		Processor ID of this XCore.
23:16	CRO		Number of the node in which this XCore is located.
15:8	CRO		XCore revision.
7:0	CRO		XCore version.

0x00:
Device
identification

C.2   xCORE Tile description 1: 0x01

This register describes the number of logical cores, synchronisers, locks and channel ends available on this xCORE tile.

Bits	Perm	Init	Description
31:24	CRO		Number of channel ends.
23:16	CRO		Number of the locks.
15:8	CRO		Number of synchronisers.
7:0	RO	-	Reserved

0x01:
xCORE Tile
description 1

C.3   xCORE Tile description 2: 0x02

This register describes the number of timers and clock blocks available on this xCORE tile.

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:8	CRO		Number of clock blocks.
7:0	CRO		Number of timers.

0x02:
xCORE Tile
description 2

C.4   Control PSwitch permissions to debug registers: 0x04

This register can be used to control whether the debug registers (marked with permission CRW) are accessible through the tile configuration registers. When this bit is set, write -access to those registers is disabled, preventing debugging of the xCORE tile over the interconnect.

Bits	Perm	Init	Description
31	CRW	0	When 1 the PSwitch is restricted to RO access to all CRW registers from SSwitch, XCore(PS_DBG_Scratch) and JTAG
30:1	RO	-	Reserved
0	CRW	0	When 1 the PSwitch is restricted to RO access to all CRW registers from SSwitch

0x04: Control PSwitch permissions to debug registers

C.5   Cause debug interrupts: 0x05

This register can be used to raise a debug interrupt in this xCORE tile.

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	CRW	0	1 when the processor is in debug mode.
0	CRW	0	Request a debug interrupt on the processor.

0x05: Cause debug interrupts

C.6   xCORE Tile clock divider: 0x06

This register contains the value used to divide the PLL clock to create the xCORE tile clock. The divider is enabled under control of the tile control register

Bits	Perm	Init	Description
31	CRW	0	Clock disable. Writing '1' will remove the clock to the tile.
30:16	RO	-	Reserved
15:0	CRW	0	Clock divider.

0x06: xCORE Tile clock divider

C.7   Security configuration: 0x07

Copy of the security register as read from OTP.

Bits	Perm	Init	Description
31	CRO		Disables write permission on this register
30:15	RO	-	Reserved
14	CRO		Disable access to XCore's global debug
13	RO	-	Reserved
12	CRO		lock all OTP sectors
11:8	CRO		lock bit for each OTP sector
7	CRO		Enable OTP reduanacy
6	RO	-	Reserved
5	CRO		Override boot mode and read boot image from OTP
4	CRO		Disable JTAG access to the PLL/BOOT configuration registers
3:1	RO	-	Reserved
0	CRO		Disable access to XCore's JTAG debug TAP

0x07:
Security
configuration

C.8 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over the switch. This is the same set of registers as the Debug Scratch registers in the processor status.

0x20 .. 0x27:
Debug
scratch

Bits	Perm	Init	Description
31:0	CRW		Value.

C.9 PC of logical core 0: 0x40

Value of the PC of logical core 0.

0x40:
PC of logical
core 0

Bits	Perm	Init	Description
31:0	CRO		Value.

C.10 PC of logical core 1: 0x41

Value of the PC of logical core 1.

<table>
<tr><td rowspan="2">**0x41:**
PC of logical
core 1</td><td>**Bits**</td><td>**Perm**</td><td>**Init**</td><td>**Description**</td></tr>
<tr><td>31:0</td><td>CRO</td><td></td><td>Value.</td></tr>
</table>

C.11 PC of logical core 2: 0x42

Value of the PC of logical core 2.

<table>
<tr><td rowspan="2">**0x42:**
PC of logical
core 2</td><td>**Bits**</td><td>**Perm**</td><td>**Init**</td><td>**Description**</td></tr>
<tr><td>31:0</td><td>CRO</td><td></td><td>Value.</td></tr>
</table>

C.12 PC of logical core 3: 0x43

Value of the PC of logical core 3.

<table>
<tr><td rowspan="2">**0x43:**
PC of logical
core 3</td><td>**Bits**</td><td>**Perm**</td><td>**Init**</td><td>**Description**</td></tr>
<tr><td>31:0</td><td>CRO</td><td></td><td>Value.</td></tr>
</table>

C.13 PC of logical core 4: 0x44

Value of the PC of logical core 4.

<table>
<tr><td rowspan="2">**0x44:**
PC of logical
core 4</td><td>**Bits**</td><td>**Perm**</td><td>**Init**</td><td>**Description**</td></tr>
<tr><td>31:0</td><td>CRO</td><td></td><td>Value.</td></tr>
</table>

C.14 PC of logical core 5: 0x45

Value of the PC of logical core 5.

<table>
<tr><td rowspan="2">**0x45:**
PC of logical
core 5</td><td>**Bits**</td><td>**Perm**</td><td>**Init**</td><td>**Description**</td></tr>
<tr><td>31:0</td><td>CRO</td><td></td><td>Value.</td></tr>
</table>

PRELIMINARY

C.15   PC of logical core 6: 0x46

Value of the PC of logical core 6.

0x46:
PC of logical
core 6

Bits	Perm	Init	Description
31:0	CRO		Value.

C.16   PC of logical core 7: 0x47

Value of the PC of logical core 7.

0x47:
PC of logical
core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

C.17   SR of logical core 0: 0x60

Value of the SR of logical core 0

0x60:
SR of logical
core 0

Bits	Perm	Init	Description
31:0	CRO		Value.

C.18   SR of logical core 1: 0x61

Value of the SR of logical core 1

0x61:
SR of logical
core 1

Bits	Perm	Init	Description
31:0	CRO		Value.

C.19   SR of logical core 2: 0x62

Value of the SR of logical core 2

0x62:
SR of logical
core 2

Bits	Perm	Init	Description
31:0	CRO		Value.

C.20   SR of logical core 3: 0x63

Value of the SR of logical core 3

0x63:
SR of logical
core 3

Bits	Perm	Init	Description
31:0	CRO		Value.

C.21   SR of logical core 4: 0x64

Value of the SR of logical core 4

0x64:
SR of logical
core 4

Bits	Perm	Init	Description
31:0	CRO		Value.

C.22   SR of logical core 5: 0x65

Value of the SR of logical core 5

0x65:
SR of logical
core 5

Bits	Perm	Init	Description
31:0	CRO		Value.

C.23   SR of logical core 6: 0x66

Value of the SR of logical core 6

0x66:
SR of logical
core 6

Bits	Perm	Init	Description
31:0	CRO		Value.

C.24   SR of logical core 7: 0x67

Value of the SR of logical core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

D Node Configuration

The digital node control registers can be accessed using configuration reads and writes (use `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ↪ ...)` for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	System switch description
0x04	RW	Switch configuration
0x05	RW	Switch node identifier
0x06	RW	PLL settings
0x07	RW	System switch clock divider
0x08	RW	Reference clock
0x09	R	System JTAG device ID register
0x0A	R	System USERCODE register
0x0C	RW	Directions 0-7
0x0D	RW	Directions 8-15
0x10	RW	DEBUG_N configuration, tile 0
0x11	RW	DEBUG_N configuration, tile 1
0x1F	RO	Debug source
0x20 .. 0x28	RW	Link status, direction, and network
0x40 .. 0x47	RO	PLink status and network
0x80 .. 0x88	RW	Link configuration and initialization
0xA0 .. 0xA7	RW	Static link configuration

Figure 33: Summary

D.1 Device identification: 0x00

This register contains version and revision identifiers and the mode-pins as sampled at boot-time.

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Sampled values of BootCtl pins on Power On Reset.
15:8	RO		SSwitch revision.
7:0	RO		SSwitch version.

0x00: Device identification

D.2 System switch description: 0x01

This register specifies the number of processors and links that are connected to this switch.

0x01:
System
switch
description

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Number of SLinks on the SSwitch.
15:8	RO		Number of processors on the SSwitch.
7:0	RO		Number of processors on the device.

D.3 Switch configuration: 0x04

This register enables the setting of two security modes (that disable updates to the PLL or any other registers) and the header-mode.

0x04:
Switch
configuration

Bits	Perm	Init	Description
31	RW	0	0 = SSCTL registers have write access. 1 = SSCTL registers can not be written to.
30:9	RO	-	Reserved
8	RW	0	0 = PLL_CTL_REG has write access. 1 = PLL_CTL_REG can not be written to.
7:1	RO	-	Reserved
0	RW	0	0 = 2-byte headers, 1 = 1-byte headers (reset as 0).

D.4 Switch node identifier: 0x05

This register contains the node identifier.

0x05:
Switch node
identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	The unique ID of this node.

D.5 PLL settings: 0x06

An on-chip PLL multiplies the input clock up to a higher frequency clock, used to clock the I/O, processor, and switch, see Oscillator. Note: a write to this register will cause the tile to be reset.

	Bits	Perm	Init	Description
	31	RW		If set to 1, the chip will not be reset
	30	RW		If set to 1, the chip will not wait for the PLL to re-lock. Only use this if a gradual change is made to the PLL
	29	DW		If set to 1, set the PLL to be bypassed
	28	DW		If set to 1, set the boot mode to boot from JTAG
	27:26	RO	-	Reserved
	25:23	RW		Output divider value range from 1 (8'h0) to 250 (8'hF9). P value.
	22:21	RO	-	Reserved
	20:8	RW		Feedback multiplication ratio, range from 1 (8'h0) to 255 (8'hFE). M value.
	7	RO	-	Reserved
0x06: PLL settings	6:0	RW		Oscilator input divider value range from 1 (8'h0) to 32 (8'h0F). N value.

D.6 System switch clock divider: 0x07

Sets the ratio of the PLL clock and the switch clock.

	Bits	Perm	Init	Description
0x07: System switch clock divider	31:16	RO	-	Reserved
	15:0	RW	0	SSwitch clock generation

D.7 Reference clock: 0x08

Sets the ratio of the PLL clock and the reference clock used by the node.

	Bits	Perm	Init	Description
0x08: Reference clock	31:16	RO	-	Reserved
	15:0	RW	3	Software ref. clock divider

D.8   System JTAG device ID register: 0x09

0x09:
System JTAG
device ID
register

Bits	Perm	Init	Description
31:28	RO		
27:12	RO		
11:1	RO		
0	RO		

D.9   System USERCODE register: 0x0A

0x0A:
System
USERCODE
register

Bits	Perm	Init	Description
31:18	RO		JTAG USERCODE value programmed into OTP SR
17:0	RO		metal fixable ID code

D.10   Directions 0-7: 0x0C

This register contains eight directions, for packets with a mismatch in bits 7..0 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

0x0C:
Directions
0-7

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose dimension is 7.
27:24	RW	0	The direction for packets whose dimension is 6.
23:20	RW	0	The direction for packets whose dimension is 5.
19:16	RW	0	The direction for packets whose dimension is 4.
15:12	RW	0	The direction for packets whose dimension is 3.
11:8	RW	0	The direction for packets whose dimension is 2.
7:4	RW	0	The direction for packets whose dimension is 1.
3:0	RW	0	The direction for packets whose dimension is 0.

D.11   Directions 8-15: 0x0D

This register contains eight directions, for packets with a mismatch in bits 15..8 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

PRELIMINARY

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose dimension is F.
27:24	RW	0	The direction for packets whose dimension is E.
23:20	RW	0	The direction for packets whose dimension is D.
19:16	RW	0	The direction for packets whose dimension is C.
15:12	RW	0	The direction for packets whose dimension is B.
11:8	RW	0	The direction for packets whose dimension is A.
7:4	RW	0	The direction for packets whose dimension is 9.
3:0	RW	0	The direction for packets whose dimension is 8.

0x0D:
Directions
8-15

D.12 DEBUG_N configuration, tile 0: 0x10

Configures the behavior of the DEBUG_N pin.

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set 1 to enable GlobalDebug to generate debug request to XCore.
0	RW	0	Set 1 to enable inDebug bit to drive GlobalDebug.

0x10:
DEBUG_N con-
figuration,
tile 0

D.13 DEBUG_N configuration, tile 1: 0x11

Configures the behavior of the DEBUG_N pin.

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set 1 to enable GlobalDebug to generate debug request to XCore.
0	RW	0	Set 1 to enable inDebug bit to drive GlobalDebug.

0x11:
DEBUG_N con-
figuration,
tile 1

D.14 Debug source: 0x1F

Contains the source of the most recent debug event.

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		If set, external pin, is the source of last GlobalDebug event.
3:2	RO	-	Reserved
1	RW		If set, XCore1 is the source of last GlobalDebug event.
0	RW		If set, XCore0 is the source of last GlobalDebug event.

0x1F:
Debug source

D.15 Link status, direction, and network: 0x20 .. 0x28

These registers contain status information for low level debugging (read-only), the network number that each link belongs to, and the direction that each link is part of. The registers control links 0..7.

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		Identify the SRC_TARGET type 0 - SLink, 1 - PLink, 2 - SSCTL, 3 - Undefine.
23:16	RO		When the link is in use, this is the destination link number to which all packets are sent.
15:12	RO	-	Reserved
11:8	RW	0	The direction that this link operates in.
7:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, reset as 0.
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO		1 when the dest side of the link is in use.
0	RO		1 when the source side of the link is in use.

0x20 .. 0x28:
Link status,
direction, and
network

D.16 PLink status and network: 0x40 .. 0x47

These registers contain status information and the network number that each processor-link belongs to.

PRELIMINARY

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		Identify the SRC_TARGET type 0 - SLink, 1 - PLink, 2 - SSCTL, 3 - Undefine.
23:16	RO		When the link is in use, this is the destination link number to which all packets are sent.
15:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, reset as 0.
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO		1 when the dest side of the link is in use.
0	RO		1 when the source side of the link is in use.

0x40 .. 0x47:
PLink status
and network

D.17 Link configuration and initialization: 0x80 .. 0x88

These registers contain configuration and debugging information specific to external links. The link speed and width can be set, the link can be initialized, and the link status can be monitored. The registers control links 0..7.

Bits	Perm	Init	Description
31	RW		Write to this bit with '1' will enable the XLink, writing '0' will disable it. This bit controls the muxing of ports with overlapping xlinks.
30	RW	0	0: operate in 2 wire mode; 1: operate in 5 wire mode
29:28	RO	-	Reserved
27	RO		Rx buffer overflow or illegal token encoding received.
26	RO	0	This end of the xlink has issued credit to allow the remote end to transmit
25	RO	0	This end of the xlink has credit to allow it to transmit.
24	WO		Clear this end of the xlink's credit and issue a HELLO token.
23	WO		Reset the receiver. The next symbol that is detected will be the first symbol in a token.
22	RO	-	Reserved
21:11	RW	0	Specify min. number of idle system clocks between two continuous symbols witin a transmit token -1.
10:0	RW	0	Specify min. number of idle system clocks between two continuous transmit tokens -1.

0x80 .. 0x88:
Link
configuration
and
initialization

PRELIMINARY

D.18 Static link configuration: 0xA0 .. 0xA7

These registers are used for static (ie, non-routed) links. When a link is made static, all traffic is forwarded to the designated channel end and no routing is attempted. The registers control links C, D, A, B, G, H, E, and F in that order.

0xA0 .. 0xA7:
Static link
configuration

Bits	Perm	Init	Description
31	RW	0	Enable static forwarding.
30:5	RO	-	Reserved
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.

E Device Errata

This section describes minor operational differences from the data sheet and recommended workarounds. As device and documentation issues become known, this section will be updated the document revised.

To guarantee a logic low is seen on the pins RST_N, TRST_N, TMS, and TDI, the driving circuit should present an impedance of less than 100 Ω to ground. Usually this is not a problem for CMOS drivers driving single inputs. If one or more of these inputs are placed in parallel, however, additional logic buffers may be required to guarantee correct operation.

For static inputs tied high or low, the relevant input pin should be tied directly to GND or VDDIO.

F JTAG, xSCOPE and Debugging

If you intend to design a board that can be used with the XMOS toolchain and xTAG debugger, you will need an xSYS header on your board. Figure 34 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.
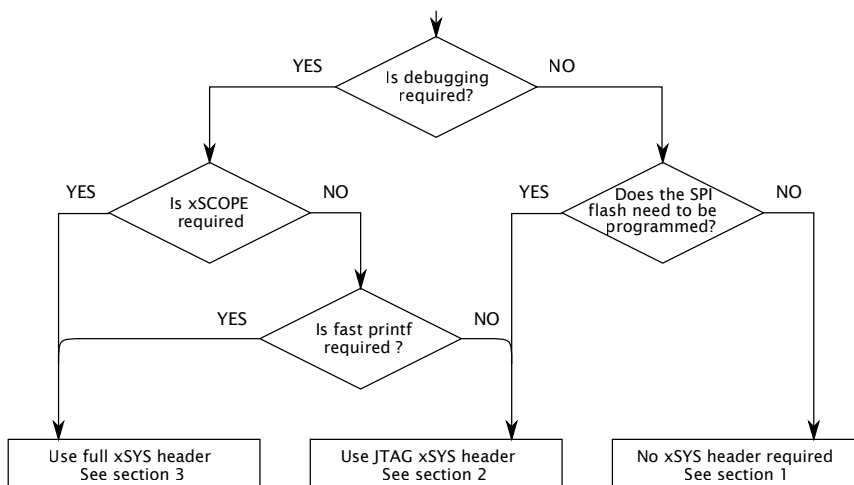
Figure 34:
Decision diagram for the xSYS header

F.1 No xSYS header

The use of an xSYS header is optional, and may not be required for volume production designs. However, the XMOS toolchain expects the xSYS header; if you do not have an xSYS header then you must provide your own method for writing to flash/OTP and for debugging.

PRELIMINARY

F.2   JTAG-only xSYS header

The xSYS header connects to an xTAG debugger, which has a 20-pin 0.1" female IDC header. The design will hence need a male IDC header. We advise to use a boxed header to guard against incorrect plug-ins. If you use a 90 degree angled header, make sure that pins 2, 4, 6, ..., 20 are along the edge of the PCB.

Connect pins 4, 8, 12, 16, 20 of the xSYS header to ground, and then connect:

▶ TDI to pin 5 of the xSYS header

▶ TMS to pin 7 of the xSYS header

▶ TCK to pin 9 of the xSYS header

▶ DEBUG_N to pin 11 of the xSYS header

▶ TDO to pin 13 of the xSYS header

The RST_N net should be open-drain, active-low, and have a pull-up to VDDIO.

F.3   Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section F.2, and then connect a 2-wire xCONNECT Link to the xSYS header. The links can be found in the Signal description table (Section 4): they are labelled XL0, XL1, etc in the function column. The 2-wire link comprises two inputs and outputs, labelled $_{out}^{1}$, $_{out}^{0}$, $_{in}^{0}$, and $_{in}^{1}$. For example, if you choose to use XL0 for xSCOPE I/O, you need to connect up $XL0_{out}^{1}$, $XL0_{out}^{0}$, $XL0_{in}^{0}$, $XL0_{in}^{1}$ as follows:

▶ $XL0_{out}^{1}$ (X0D43) to pin 6 of the xSYS header with a 33R series resistor close to the device.

▶ $XL0_{out}^{0}$ (X0D42) to pin 10 of the xSYS header with a 33R series resistor close to the device.

▶ $XL0_{in}^{0}$ (X0D41) to pin 14 of the xSYS header.

▶ $XL0_{in}^{1}$ (X0D40) to pin 18 of the xSYS header.

G   Schematics Design Check List

☑   This section is a checklist for use by schematics designers using the XL212-512-TQ128. Each of the following sections contains items to check for each design.

G.1   Power supplies

☐   VDDIO and OTP_VCC supply is within specification before the VDD (core) supply is turned on. Specifically, the VDDIO and OTP_VCC supply is within specification before VDD (core) reaches 0.4V (Section 11).

☐   The VDD (core) supply ramps monotonically (rises constantly) from 0V to its final value (0.95V - 1.05V) within 10ms (Section 11).

☐   The VDD (core) supply is capable of supplying 600mA (Section 11).

☐   PLL_AVDD is filtered with a low pass filter, for example an RC filter, see Section 11   .

G.2   Power supply decoupling

☐   The design has multiple decoupling capacitors per supply, for example at least four 0402 or 0603 size surface mount capacitors of 100nF in value, per supply (Section 11).

☐   A bulk decoupling capacitor of at least 10uF is placed on each supply (Section 11).

G.3   Power on reset

☐   The RST_N and TRST_N pins are asserted (low) during or after power up. The device is not used until these resets have taken place.
As the errata in the datasheets show, the internal pull-ups on these two pins can occasionally provide stronger than normal pull-up currents. For this reason, an RC type reset circuit is discouraged as behavior would be unpredictable. A voltage supervisor type reset device is recommended to guarantee a good reset. This also has the benefit of resetting the system should the relevant supply go out of specification.

G.4   Clock

☐   The CLK input pin is supplied with a clock with monotonic rising edges and low jitter.

☐ You have chosen an input clock frequency that is supported by the device (Section 7).

G.5 Boot

☐ The device is connected to a QSPI flash for booting, connected to X0D01, X0D04..X0D07, and X0D10 (Section 8). If not, you must boot the device through OTP or JTAG, or set it to boot from SPI and connect a SPI flash.

☐ The Flash that you have chosen is supported by **xflash**, or you have created a specification file for it.

G.6 JTAG, XScope, and debugging

☐ You have decided as to whether you need an XSYS header or not (Section F)

☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section F).

G.7 GPIO

☐ You have not mapped both inputs and outputs to the same multi-bit port.

☐ Pins X0D04, X0D05, X0D06, and X0D07 are output only and are, after reset, pulled high and low appropriately (Section 8)

G.8 Multi device designs

Skip this section if your design only includes a single XMOS device.

☐ One device is connected to a SPI flash for booting.

☐ Devices that boot from link have MODE2 grounded and MODE3 NC. These device must have link XLB connected to a device to boot from (see 8).

☐ If you included an XSYS header, you have included buffers for RST_N, TRST_N, TMS, TCK, MODE2, and MODE3 (Section E).

PRELIMINARY

H   PCB Layout Design Check List

☑   This section is a checklist for use by PCB designers using the XS2-L12A-512-TQ128. Each of the following sections contains items to check for each design.

H.1   Ground Plane

☐   Multiple vias (eg, 9) have been used to connect the center pad to the PCB ground plane. These minimize impedance and conduct heat away from the device. (Section 11.2).

☐   Other than ground vias, there are no (or only a few) vias underneath or closely around the device. This create a good, solid, ground plane.

H.2   Power supply decoupling

☐   The decoupling capacitors are all placed close to a supply pin (Section 11).

☐   The decoupling capacitors are spaced around the device (Section 11).

☐   The ground side of each decoupling capacitor has a direct path back to the center ground of the device.

H.3   PLL_AVDD

☐   The PLL_AVDD filter (especially the capacitor) is placed close to the PLL_AVDD pin (Section 11).

I Associated Design Documentation

Document Title	Information	Document Number
Estimating Power Consumption For XS1-L Devices	Power consumption	X4271
Programming XC on XMOS Devices	Timers, ports, clocks, cores and channels	X9577
xTIMEcomposer User Guide	Compilers, assembler and linker/mapper Timing analyzer, xScope, debugger Flash and OTP programming utilities	X3766

J Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	X7879
XS1 Port I/O Timing	Port timings	X5821
xCONNECT Architecture	Link, switch and system information	X4249
XS1-L Link Performance and Design Guidelines	Link timings	X2999
XS1-L Clock Frequency Control	Advanced clock control	X1433
XS1-L Active Power Conservation	Low-power mode during idle	X7411

PRELIMINARY

K   Revision History

Date	Description
2015-03-20	Preliminary release
2015-04-14	Added RST to pins to be pulled hard, and removed reference to TCK from Errata
	Removed TRST_N references in packages that have no TRST_N
2015-04-29	VDDIOR and VDD (pins 47/48) switched - Section 3

XMOS®

Copyright © 2015, All Rights Reserved.