



Welcome to [E-XFL.COM](https://www.e-xfl.com)

What is "[Embedded - Microcontrollers](#)"?

"[Embedded - Microcontrollers](#)" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "[Embedded - Microcontrollers](#)"

Details

Product Status	Obsolete
Core Processor	XCore
Core Size	32-Bit 10-Core
Speed	2000MIPS
Connectivity	-
Peripherals	-
Number of I/O	88
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	128-TQFP Exposed Pad
Supplier Device Package	128-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xlf210-512-tq128-c20

- ▶ **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](#)
- ▶ **Flash** The device has a built-in 2MBflash. Section [8](#)
- ▶ **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section [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 XLF210-512-TQ128 Features

► Multicore Microcontroller with Advanced Multi-Core RISC Architecture

- 10 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 $\frac{1}{5}$ and $\frac{1}{5}$ 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 ends (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
- 2MB internal flash for application code and overlays

► 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

Signal	Function	Type	Properties
X0D41	$X_0L0_{in}^0$ 8D ⁵ 16B ¹³	I/O	IOL, PD
X0D42	$X_0L0_{out}^0$ 8D ⁶ 16B ¹⁴	I/O	IOL, PD
X0D43	$X_0L0_{out}^0$ 8D ⁷ 16B ¹⁵	I/O	IOL, PD
X1D00	$X_0L7_{in}^0$ 1A ⁰	I/O	IOR, PD
X1D01	$X_0L7_{in}^1$ 1B ⁰	I/O	IOR, PD
X1D02	$X_0L4_{in}^0$ 4A ⁰ 8A ⁰ 16A ⁰ 32A ²⁰	I/O	IOR, PD
X1D03	$X_0L4_{out}^0$ 4A ¹ 8A ¹ 16A ¹ 32A ²¹	I/O	IOR, PD
X1D04	$X_0L4_{out}^1$ 4B ⁰ 8A ² 16A ² 32A ²²	I/O	IOR, PD
X1D05	$X_0L4_{out}^2$ 4B ¹ 8A ³ 16A ³ 32A ²³	I/O	IOR, PD
X1D06	$X_0L4_{out}^3$ 4B ² 8A ⁴ 16A ⁴ 32A ²⁴	I/O	IOR, PD
X1D07	$X_0L4_{out}^4$ 4B ³ 8A ⁵ 16A ⁵ 32A ²⁵	I/O	IOR, PD
X1D08	$X_0L7_{in}^4$ 4A ² 8A ⁶ 16A ⁶ 32A ²⁶	I/O	IOR, PD
X1D09	$X_0L7_{in}^3$ 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	$X_0L3_{in}^1$ 4D ⁰ 8B ² 16A ¹⁰	I/O	IOL, PD
X1D17	$X_0L3_{in}^0$ 4D ¹ 8B ³ 16A ¹¹	I/O	IOL, PD
X1D18	$X_0L3_{out}^0$ 4D ² 8B ⁴ 16A ¹²	I/O	IOL, PD
X1D19	$X_0L3_{out}^1$ 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	$X_0L3_{out}^4$ 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	$X_0L0_{out}^2$ 1K ⁰	I/O	IOL, PD
X1D35	$X_0L0_{out}^3$ 1L ⁰	I/O	IOL, PD
X1D36	$X_0L0_{out}^4$ 1M ⁰ 8D ⁰ 16B ⁸	I/O	IOL, PD
X1D37	$X_0L3_{in}^4$ 1N ⁰ 8D ¹ 16B ⁹	I/O	IOL, PD
X1D38	$X_0L3_{in}^3$ 1O ⁰ 8D ² 16B ¹⁰	I/O	IOL, PD
X1D39	$X_0L3_{in}^2$ 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

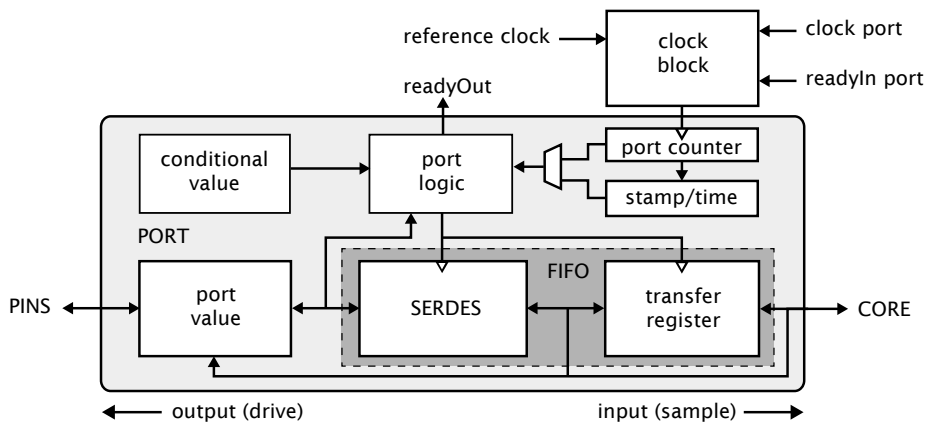


Figure 4:
Port block
diagram

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.

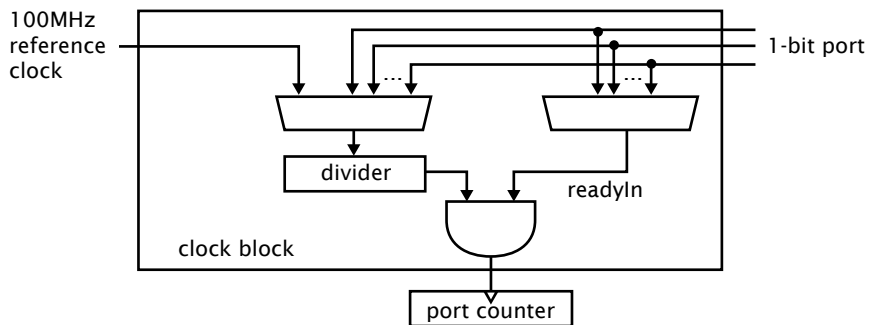
The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for 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.

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.

Figure 5:
Clock block
diagram



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

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

6.5 Channels and Channel Ends

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

6.6 xCONNECT Switch and Links

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

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

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

8 Boot Procedure

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

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

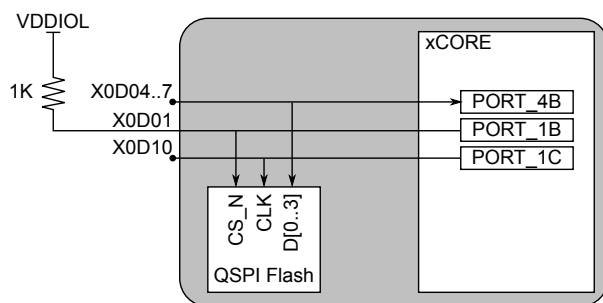


Figure 8:
QSPI port
connectivity

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

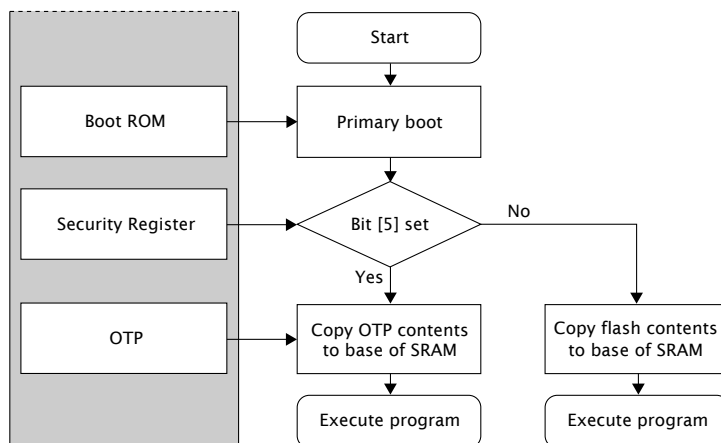


Figure 9:
Boot
procedure

The boot image has the following format:

- A 32-bit program size s in words.

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

12.6 Clock

Figure 20:
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			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-LF Clock Frequency Control document,

12.7 xCORE Tile I/O AC Characteristics

Figure 21:
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

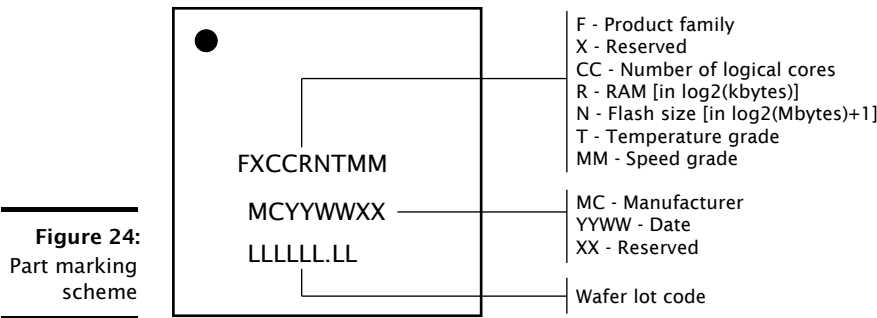
Figure 22:
Link performance

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.

13.1 Part Marking



14 Ordering Information

Figure 25:
Orderable
part numbers

Product Code	Marking	Qualification	Speed Grade
XLF210-512-TQ128-C20	L11092C20	Commercial	1000 MIPS
XLF210-512-TQ128-I20	L11092I20	Industrial	1000 MIPS

B.4 xCORE Tile boot status: 0x03

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

0x03:
xCORE Tile
boot status

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.

B.5 Security configuration: 0x05

Copy of the security register as read from OTP.

0x05:
Security
configuration

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 redundancy
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

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.

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 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.

0x00:
Device
identification

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.

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.

0x01:
xCORE Tile
description 1

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

C.3 xCORE Tile description 2: 0x02

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

0x02:
xCORE Tile
description 2

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

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.

0x04: Control PSwitch permissions to debug registers	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

C.5 Cause debug interrupts: 0x05

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

0x05: Cause debug interrupts	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.

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](#)

0x06: xCORE Tile clock divider	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.

C.7 Security configuration: 0x07

Copy of the security register as read from OTP.

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

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	Reserved
0x11	RW	Reserved.
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 29:
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.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 governed 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 governed by the most significant mismatching bit.

0x1F:
Debug source

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		Reserved.
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.

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.

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

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.

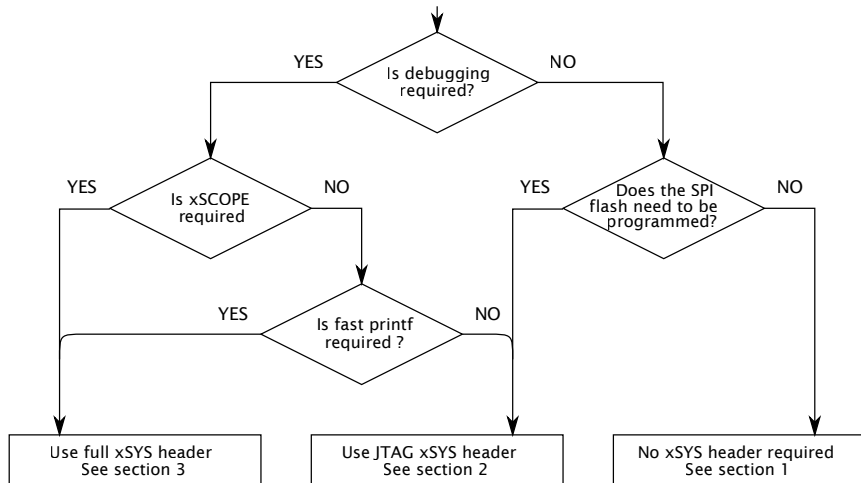
D.16 PLink status and network: 0x40 .. 0x47

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

E 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 30 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

Figure 30:
Decision
diagram for
the xSYS
header



E.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.

E.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
- ▶ TDO to pin 13 of the xSYS header

F.5 Boot

- ☐ X0D01 has a 1K pull-up to VDDIOL (Section 8).
- ☐ The device is kept in reset for at least 1 ms after VDDIOL has reached its minimum level (Section 8).

F.6 JTAG, XScope, and debugging

- ☐ You have decided as to whether you need an XSYS header or not (Section E)
- ☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section E).

F.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, during and after reset, pulled low or not connected (Section 8)

F.8 Multi device designs

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

- ☐ One device is connected to a QSPI or SPI flash for booting.
- ☐ Devices that boot from link have, for example, X0D06 pulled high and have link XL0 connected to a device to boot from (Section 8).