

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	Active
Core Processor	XCore
Core Size	32-Bit Quad-Core
Speed	400MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	28
Program Memory Size	64KB (16K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
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	48-TQFP Exposed Pad
Supplier Device Package	48-TQFP (7x7)
Purchase URL	<a href="https://www.e-xfl.com/product-detail/xmos/xs1-l4a-64-tq48-c4">https://www.e-xfl.com/product-detail/xmos/xs1-l4a-64-tq48-c4</a>

## Table of Contents

1	xCORE Multicore Microcontrollers	2
2	XS1-L4A-64-TQ48 Features	4
3	Pin Configuration	5
4	Signal Description	6
5	Product Overview	8
6	PLL	11
7	Boot Procedure	12
8	Memory	14
9	JTAG	15
10	Board Integration	16
11	DC and Switching Characteristics	20
12	Package Information	23
13	Ordering Information	24
	Appendices	25
A	Configuration of the XS1	25
B	Processor Status Configuration	27
C	Tile Configuration	36
D	Node Configuration	42
E	XMOS USB Interface	49
F	Device Errata	49
G	JTAG, xSCOPE and Debugging	49
H	Schematics Design Check List	52
I	PCB Layout Design Check List	55
J	Associated Design Documentation	56
K	Related Documentation	56
L	Revision History	57

## TO OUR VALUED CUSTOMERS

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

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

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

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.

## 7.1 Boot from SPI master

If set to boot from SPI master, the processor enables the four pins specified in Figure 9, and drives the SPI clock at 2.5 MHz (assuming a 400 MHz core clock). A READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

**Figure 9:**  
SPI master  
pins

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

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

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

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

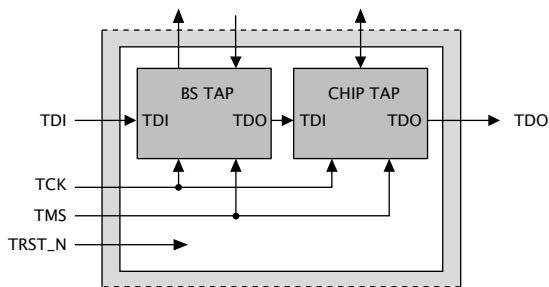
## 7.2 Boot from xConnect Link

If set to boot from an xConnect Link, the processor enables Link B around 200 ns after the boot process starts. Enabling the Link switches off the pull-down on resistors X0D16..X0D19, drives X0D16 and X0D17 low (the initial state for the Link), and monitors pins X0D18 and X0D19 for boot-traffic. X0D18 and X0D19 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.

**Figure 11:**  
JTAG chain structure



The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 12.

**Figure 12:**  
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	0	0	1	1	0	0	0	1	1	0	0	1	1
0				0				0				0				2				6				3				3				

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 13. The OTP User ID field is read from bits [22:31] of the security register , see §8.1 (all zero on unprogrammed devices).

**Figure 13:**  
USERCODE  
return value

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				2				8				0				0				0				

## 10 Board Integration

The device has the following power supply pins:

- VDD pins for the xCORE Tile
- VDDIO pins for the I/O lines
- PLL\_AVDD pins for the PLL

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 supply must ramp to its final value before VDD reaches 0.4 V.

### 10.3 Moisture Sensitivity

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

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

### 11.7 xCORE Tile I/O AC Characteristics

**Figure 22:**  
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](#).

### 11.8 xConnect Link Performance

**Figure 23:**  
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.

### 11.9 JTAG Timing

**Figure 24:**  
JTAG timing

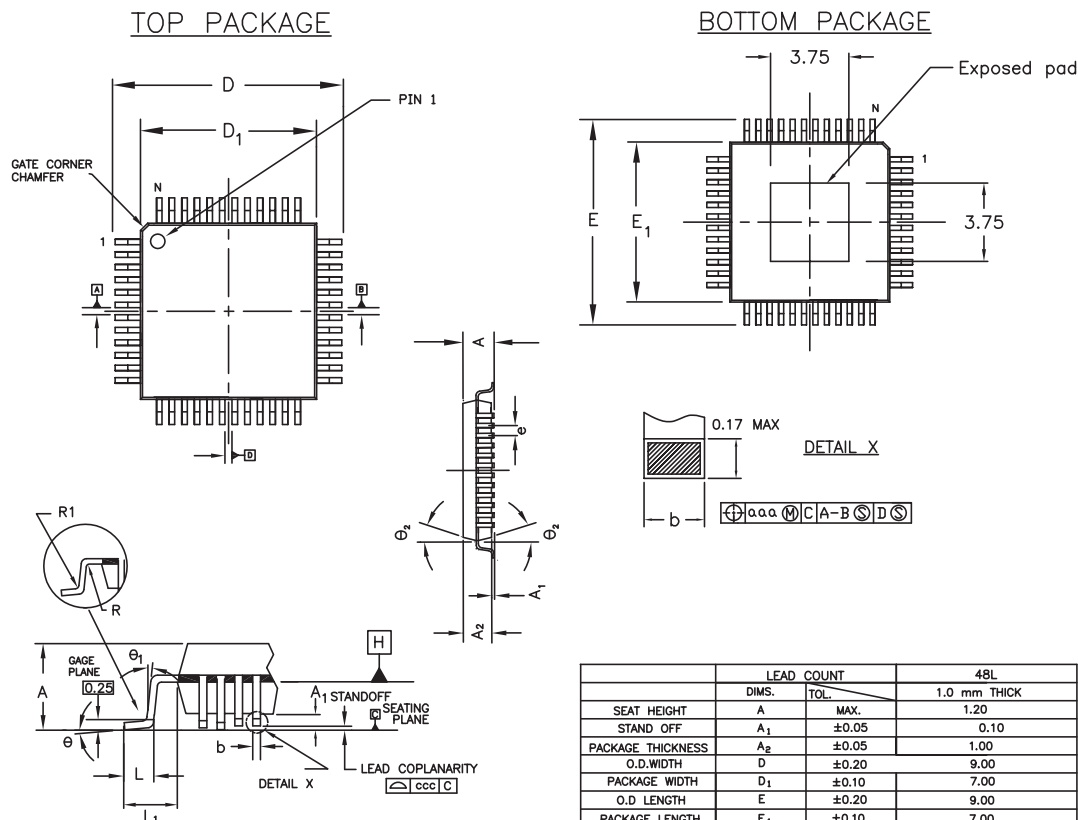
Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			18	MHz	
f(TCK_B)	TCK frequency (boundary scan)			10	MHz	
T(SETUP)	TDO to TCK setup time	5			ns	A
T(HOLD)	TDO to TCK hold time	5			ns	A
T(DELAY)	TCK to output delay			15	ns	B

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.

## 12 Package Information



### NOTES:

1. All dimensions in millimeters.
2. Dimension tolerances shall be  $\pm 0.05\text{mm}$  unless otherwise specified.
3. Foot length 'L' is measured at gage plane 0.25mm above seating plane.
4. Dimension D<sub>1</sub> and E<sub>1</sub> are bottom package width and length and are measured at datum plane H.
5. Leadframe material: Eftec 64T Cu (or equivalent), 0.127mm (0.005") thick.

	LEAD COUNT		48L
	DIMS.	TOL.	1.0 mm THICK
SEAT HEIGHT	A	MAX.	1.20
STAND OFF	A <sub>1</sub>	$\pm 0.05$	0.10
PACKAGE THICKNESS	A <sub>2</sub>	$\pm 0.05$	1.00
O.D.WIDTH	D	$\pm 0.20$	9.00
PACKAGE WIDTH	D <sub>1</sub>	$\pm 0.10$	7.00
O.D LENGTH	E	$\pm 0.20$	9.00
PACKAGE LENGTH	E <sub>1</sub>	$\pm 0.10$	7.00
FOOT LENGTH	L	$+0.15/-0.10$	0.60
HALF FOOTPRINT	L <sub>1</sub>	REF	1.00
LEAD PITCH	e	TYPE	0.50
LEAD WIDTH	b	$\pm 0.05$	0.22
FOOT ANGLE	θ		0°-7°
VERTICAL ANGLE	θ <sub>1</sub>		0° MIN.
	θ <sub>2</sub>	$\pm 1^\circ$	12°
FIRST BEND	R	TYP	0.15
SECOND BEND	R1	$\pm 0.05$	0.15
	aaa	MAX.	0.08
COPLANARITY	ccc	MAX.	0.08
JEDEC REFERENCE DRAWING		MS-026	
VARIATION DESIGNATOR		ABC	

## 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	RO	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
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 28:**  
Summary



### 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	-	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.

**0x09:**  
Ring  
Oscillator  
Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	-	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.

**0x0A:**  
Ring  
Oscillator  
Value

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

### B.11 Debug SSR: 0x10

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

**0x10:**  
Debug SSR

Bits	Perm	Init	Description
31:0	RO	-	Reserved

### B.12 Debug SPC: 0x11

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

**0x15:**  
Debug  
interrupt type

Bits	Perm	Init	Description
31:18	RO	-	Reserved
17:16	DRW		If the debug interrupt was caused by a hardware breakpoint or hardware watchpoint, this field contains the number of the breakpoint or watchpoint. If multiple breakpoints or watchpoints trigger at once, the lowest number is taken.
15:8	DRW		If the debug interrupt was caused by a logical core, this field contains the number of that core. Otherwise this field is 0.
7:3	RO	-	Reserved
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.17 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 contains the resource identifier.

**0x16:**  
Debug  
interrupt data

Bits	Perm	Init	Description
31:0	DRW		Value.

### B.18 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 logical cores are stopped when not in debug mode. Every bit which is set prevents the respective logical core from running.

### B.19 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.20 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.21 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 logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	Set to 1 to cause an instruction breakpoint if the PC is not equal to the breakpoint address. By default, the breakpoint is triggered when the PC is equal to the breakpoint address.
0	DRW	0	When 1 the instruction breakpoint is enabled.

### B.22 Data watchpoint address 1: 0x50 .. 0x53

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

## 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	RO	Device identification
0x01	RO	xCORE Tile description 1
0x02	RO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	RW	xCORE Tile clock divider
0x07	RO	Security configuration
0x10 .. 0x13	RO	PLink status
0x20 .. 0x27	CRW	Debug scratch
0x40	RO	PC of logical core 0
0x41	RO	PC of logical core 1
0x42	RO	PC of logical core 2
0x43	RO	PC of logical core 3
0x60	RO	SR of logical core 0
0x61	RO	SR of logical core 1
0x62	RO	SR of logical core 2
0x63	RO	SR of logical core 3
0x80 .. 0x9F	RO	Chanend status

**Figure 29:**  
Summary

### C.1 Device identification: 0x00

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

**0x00:**  
Device  
identification

**0x10 .. 0x13:**  
PLink status

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		00 - ChannelEnd, 01 - ERROR, 10 - PSCTL, 11 - Idle.
23:16	RO		Based on SRC_TARGET_TYPE value, it represents channelEnd ID or Idle status.
15:6	RO	-	Reserved
5:4	RO		Two-bit network identifier
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

### C.9 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.10 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	RO		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	<a href="#">Device identification</a>
0x01	RO	<a href="#">System switch description</a>
0x04	RW	<a href="#">Switch configuration</a>
0x05	RW	<a href="#">Switch node identifier</a>
0x06	RW	<a href="#">PLL settings</a>
0x07	RW	<a href="#">System switch clock divider</a>
0x08	RW	<a href="#">Reference clock</a>
0x0C	RW	<a href="#">Directions 0-7</a>
0x0D	RW	<a href="#">Directions 8-15</a>
0x10	RW	<a href="#">DEBUG_N configuration</a>
0x1F	RO	<a href="#">Debug source</a>
0x20 .. 0x27	RW	<a href="#">Link status, direction, and network</a>
0x40 .. 0x43	RW	<a href="#">PLink status and network</a>
0x80 .. 0x87	RW	<a href="#">Link configuration and initialization</a>
0xA0 .. 0xA7	RW	<a href="#">Static link configuration</a>

**Figure 30:**  
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	0x00	Chip identifier.
23:16	RO		Sampled values of pins MODE0, MODE1, ... 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.

**0x0C:**  
Directions  
0-7

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

### D.9 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.

**0x0D:**  
Directions  
8-15

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose first mismatching bit is 15.
27:24	RW	0	The direction for packets whose first mismatching bit is 14.
23:20	RW	0	The direction for packets whose first mismatching bit is 13.
19:16	RW	0	The direction for packets whose first mismatching bit is 12.
15:12	RW	0	The direction for packets whose first mismatching bit is 11.
11:8	RW	0	The direction for packets whose first mismatching bit is 10.
7:4	RW	0	The direction for packets whose first mismatching bit is 9.
3:0	RW	0	The direction for packets whose first mismatching bit is 8.

### D.10 DEBUG\_N configuration: 0x10

Configures the behavior of the DEBUG\_N pin.

**0x10:**  
DEBUG\_N  
configuration

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set to 1 to enable signals on DEBUG_N to generate DCALL on the core.
0	RW	0	When set to 1, the DEBUG_N wire will be pulled down when the node enters debug mode.

### D.11 Debug source: 0x1F

Contains the source of the most recent debug event.

**0x1F:**  
Debug source

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		If set, the external DEBUG_N pin is the source of the most recent debug interrupt.
3:1	RO	-	Reserved
0	RW		If set, the xCORE Tile is the source of the most recent debug interrupt.

### D.12 Link status, direction, and network: 0x20 .. 0x27

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 C, D, A, B, G, H, E, and F in that order.

**0x20 .. 0x27:**  
Link status,  
direction, and  
network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		If this link is currently routing data into the switch, this field specifies the type of link that the data is routed to: 0: plink 1: external link 2: internal control link
23:16	RO	0	If the link is routing data into the switch, this field specifies the destination link number to which all tokens are sent.
15:12	RO	-	Reserved
11:8	RW	0	The direction that this this link is associated with; set for routing.
7:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, set for quality of service.
3	RO	-	Reserved
2	RO	0	Set to 1 if the current packet is junk and being thrown away. A packet is considered junk if, for example, it is not routable.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.



## E XMOS USB Interface

XMOS provides a low-level USB interface for connecting the device to a USB transceiver using the UTMI+ Low Pin Interface (ULPI). The ULPI signals must be connected to the pins named in Figure 31. Note also that some ports on the same tile are used internally and are not available for use when the USB driver is active (they are available otherwise).

**Figure 31:**  
ULPI signals  
provided by  
the XMOS  
USB driver

Pin	Signal
XnD12	ULPI_STEP
XnD13	ULPI_NXT
XnD14	ULPI_DATA[0]
XnD15	ULPI_DATA[1]
XnD16	ULPI_DATA[2]

Pin	Signal
XnD17	ULPI_DATA[3]
XnD18	ULPI_DATA[4]
XnD19	ULPI_DATA[5]
XnD20	ULPI_DATA[6]
XnD21	ULPI_DATA[7]

Pin	Signal
XnD22	ULPI_DIR
XnD23	ULPI_CLK
XnD37	Unavailable
XnD38	
XnD39	

## F 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, MODE[3:0], TRST\_N, TMS, TCK and TDI, the driving circuit should present an impedance of less than 100  $\Omega$  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.

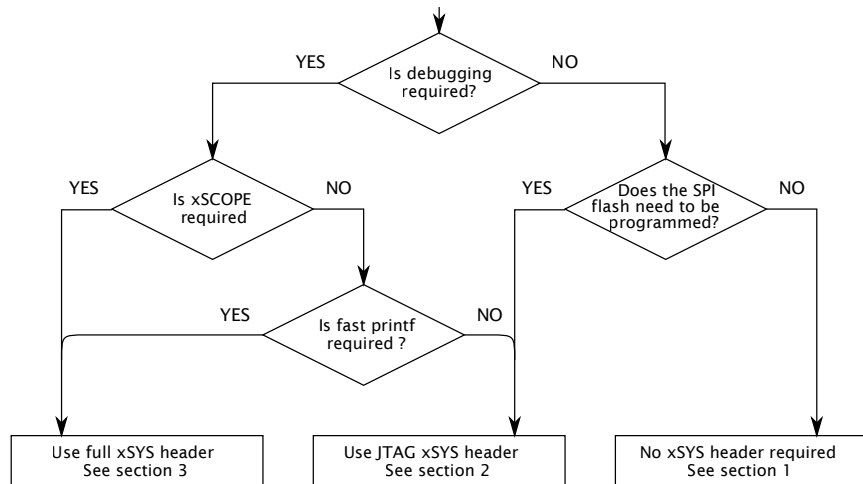
## G JTAG, xSCOPE and Debugging

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

### G.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.

**Figure 32:**  
Decision  
diagram for  
the xSYS  
header



## G.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
- ▶ RST\_N and TRST\_N to pin 15 of the xSYS header
- ▶ If MODE2 is configured high, connect MODE2 to pin 3 of the xSYS header. Do not connect to VDDIO.
- ▶ If MODE3 is configured high, connect MODE3 to pin 3 of the xSYS header. Do not connect to VDDIO.

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

## H Schematics Design Check List

- ✓ This section is a checklist for use by schematics designers using the XS1-L4A-64-TQ48. Each of the following sections contains items to check for each design.

### H.1 Power supplies

- ☐ VDDIO supply is within specification before the VDD (core) supply is turned on. Specifically, the VDDIO supply is within specification before VDD (core) reaches 0.4V (Section 10).
- ☐ The VDD (core) supply ramps monotonically (rises constantly) from 0V to its final value (0.95V - 1.05V) within 10ms (Section 10).
- ☐ The VDD (core) supply is capable of supplying 300mA (Section 10).
- ☐ PLL\_AVDD is filtered with a low pass filter, for example an RC filter, see Section 10

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

### H.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.

### H.4 Clock

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

## I PCB Layout Design Check List

- ✓ This section is a checklist for use by PCB designers using the XS1-L4A-64-TQ48. Each of the following sections contains items to check for each design.

### I.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 [10.2](#)).
- ☐ 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.

### I.2 Power supply decoupling

- ☐ The decoupling capacitors are all placed close to a supply pin (Section [10](#)).
- ☐ The decoupling capacitors are spaced around the device (Section [10](#)).
- ☐ The ground side of each decoupling capacitor has a direct path back to the center ground of the device.

### I.3 PLL\_AVDD

- ☐ The PLL\_AVDD filter (especially the capacitor) is placed close to the PLL\_AVDD pin (Section [10](#)).

## L Revision History

Date	Description
2013-01-30	New datasheet - revised part numbering
2013-02-26	New multicore microcontroller introduction Moved configuration sections to appendices
2013-07-19	Updated Features list with available ports and links - Section <a href="#">2</a> Simplified link bits in Signal Description - Section <a href="#">4</a> New JTAG, xSCOPE and Debugging appendix - Section <a href="#">G</a> New Schematics Design Check List - Section <a href="#">H</a> New PCB Layout Design Check List - Section <a href="#">I</a>
2013-12-09	Added Industrial Ambient Temperature - Section <a href="#">11.1</a>
2015-04-14	Updated Introduction - Section <a href="#">1</a> ; Pin Configuration - Section <a href="#">3</a> ; Signal Description - Section <a href="#">4</a>



Copyright © 2015, All Rights Reserved.

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