

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 12-Core
Speed	2000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	81
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°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/xu212-512-tq128-i20

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.

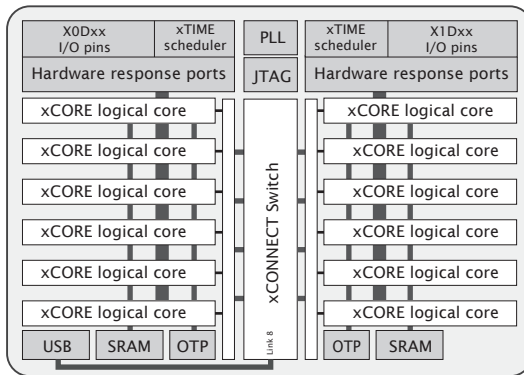


Figure 1:
XU212-512-
TQ128 block
diagram

Key features of the XU212-512-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](#)

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	$X_0L4_{out}^1$ 4A ¹ 8A ¹ 16A ¹ 32A ²¹	I/O	IOR, PD
X1D04	$X_0L4_{out}^2$ 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
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
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
X1D35	1L ⁰	I/O	IOL, PD
X1D36	1M ⁰ 8D ⁰ 16B ⁸	I/O	IOL, PD
X1D37	1N ⁰ 8D ¹ 16B ⁹	I/O	IOL, PD
X1D38	1O ⁰ 8D ² 16B ¹⁰	I/O	IOL, PD
X1D39	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

6 Product Overview

The XU212-512-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 XU212-512-TQ128, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

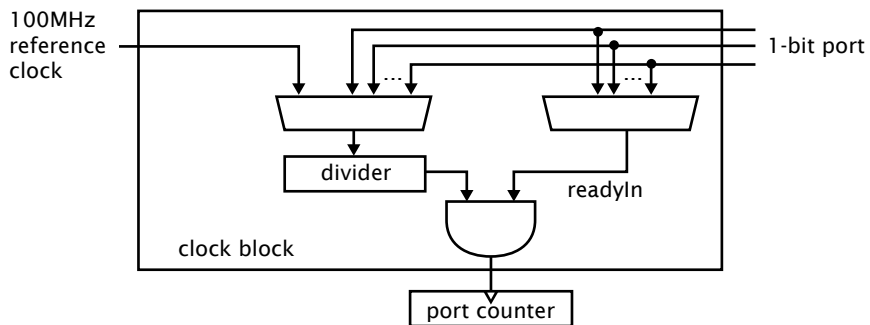


Figure 5:
Clock block
diagram

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

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

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

6.5 Channels and Channel Ends

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

6.6 xCONNECT Switch and Links

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

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

The links operate in either 2 wires per direction or 5 wires per direction mode, depending on the amount of bandwidth required. Circuit switched, streaming

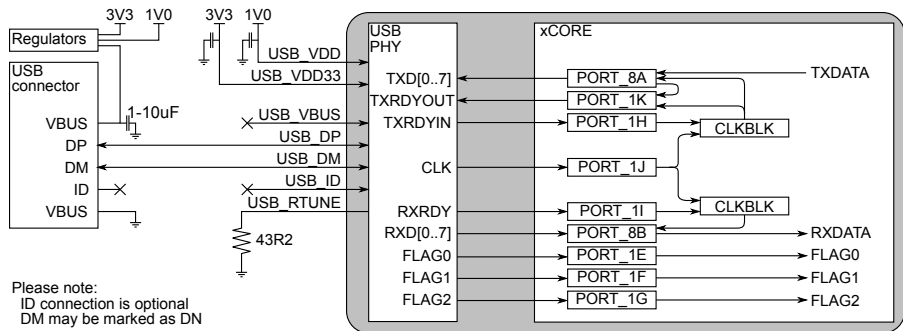


Figure 14:
Bus powered
USB-device

If you use the USB PHY to design a self-powered *USB-device*, then the device must be able detect the presence of VBus on the USB connector (so the device can disconnect its pull-up resistors from D+/D- to ensure the device does not have any voltage on the D+/D- pins when VBus is not present, “USB Back Voltage Test”). This requires USB_VBUS to be connected to the VBUS pin of the USB connector as is shown in Figure 15.

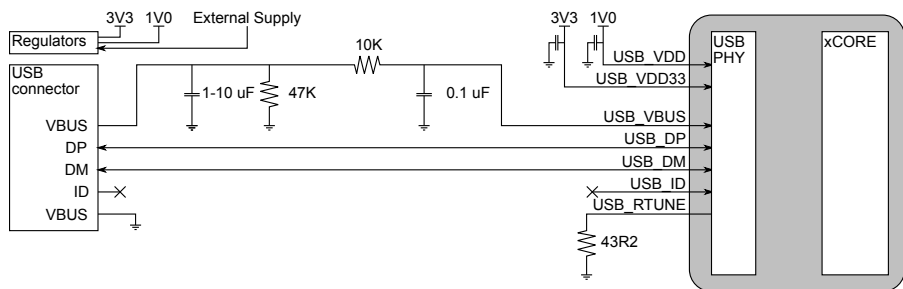


Figure 15:
Self powered
USB-device

When connecting a USB cable to the device it is possible an overvoltage transient will be present on VBus due to the inductance of the USB cable combined with the required input capacitor on VBus. The circuit in Figure 15 ensures that the transient does not damage the device. The 10k series resistor and 0.1uF capacitor ensure that any input transient is filtered and does not reach the device. The 47k resistor to ground is a bleeder resistor to discharge the input capacitor when VBus is not present. The 1-10uF input capacitor is required as part of the USB specification. A typical value would be 2.2uF to ensure the 1uF minimum requirement is met even under voltage bias conditions.

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

(for example, 100nF 0402 for each supply pin). The ground side of the decoupling capacitors should have as short a path back to the GND pins as possible. A bulk decoupling capacitor of at least 10 uF should be placed on each of these supplies.

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

12.1 USB connections

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

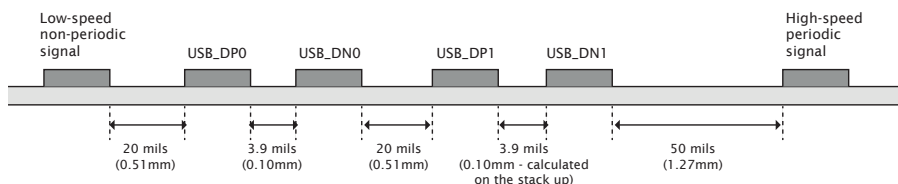
For self-powered systems, a bleeder resistor may be required to stop VBUS from floating when no USB cable is attached.

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

12.2 USB signal routing and placement

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

Figure 19:
USB trace separation showing a low speed signal, two differential pairs and a high-speed clock



12.2.1 General routing and placement guidelines

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

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS2-U12A-512-TQ128 are on the top layer) closest to GND.

- ▶ DO NOT route USB traces near clock sources, clocked circuits or magnetic devices.
- ▶ Avoid stubs on high speed USB signals.

12.3 Land patterns and solder stencils

The package is a 128 pin Thin Quad Flat Package (TQFP) with exposed ground paddle/heat slug on a 0.4mm pitch.

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

12.4 Ground and Thermal Vias

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

12.5 Moisture Sensitivity

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

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

A write message comprises the following:

control-token 36	24-bit response channel-end identifier	8-bit register number	8-bit size	data	control-token 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 37	24-bit response channel-end identifier	8-bit register number	8-bit size	control-token 1
---------------------	---	--------------------------	---------------	--------------------

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

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 contains 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.

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 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.

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.

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.

C.24 SR of logical core 7: 0x67

Value of the SR of logical core 7

0x67:
SR of logical
core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

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.

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.
30:9	RO	-	Reserved
8	RW	0	The destination processor on this node that packets received in static mode are forwarded to.
7:5	RO	-	Reserved
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.

	Bits	Perm	Init	Description
0x2C: UIFM PID	31:4	RO	-	Reserved
	3:0	RO	0	Value of the last received PID.

F.13 UIFM Endpoint: 0x30

The last endpoint seen

	Bits	Perm	Init	Description
0x30: UIFM Endpoint	31:5	RO	-	Reserved
	4	RO	0	1 if endpoint contains a valid value.
	3:0	RO	0	A copy of the last received endpoint.

F.14 UIFM Endpoint match: 0x34

This register can be used to mark UIFM endpoints as special.

	Bits	Perm	Init	Description
0x34: UIFM Endpoint match	31:16	RO	-	Reserved
	15:0	RW	0	This register contains a bit for each endpoint. If its bit is set, the endpoint will be supplied on the RX port when ORed with 0x10.

F.15 OTG Flags mask: 0x38

	Bits	Perm	Init	Description
0x38: OTG Flags mask	31:0	RW	0	Data

F.16 UIFM power signalling: 0x3C

	Bits	Perm	Init	Description
0x3C: UIFM power signalling	31:9	RO	-	Reserved
	8	RW	0	Valid
	7:0	RW	0	Data

F.17 UIFM PHY control: 0x40

Bits	Perm	Init	Description
31:19	RO	-	Reserved
18	RW	0	Set to 1 to disable pulldowns on ports 8A and 8B.
17:14	RO	-	Reserved
13	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for reset (se0). Set to 0 to clear.
12	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for resume (K). Set to 0 to clear.
11:8	RW	0	Log-2 number of clocks before any linestate change is propagated.
7	RW	0	Set to 1 to use the suspend controller handle to resume from suspend. Otherwise, the program has to poll the linestate_filt field in phy_teststatus.
6:4	RW	0	Control the the conf1,2,3 input pins of the PHY.
3:0	RO	-	Reserved

0x40:
UIFM PHY
control

H.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.

H.6 JTAG, XScope, and debugging

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

H.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 high and low appropriately (Section 8)

H.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).

I PCB Layout Design Check List

- ☒ This section is a checklist for use by PCB designers using the XS2-U12A-512-TQ128. 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 [12.4](#)).
- ☐ 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 [12](#)).
- ☐ The decoupling capacitors are spaced around the device (Section [12](#)).
- ☐ 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 [12](#)).

J Associated Design Documentation

Document Title	Information	Document Number
Estimating Power Consumption For XS1-U Devices	Power consumption	
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

K 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-U Link Performance and Design Guidelines	Link timings	
XS1-U Clock Frequency Control	Advanced clock control	