

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 16-Core
Speed	2000MIPS
Connectivity	RGMII, USB
Peripherals	-
Number of I/O	67
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	<a href="https://www.e-xfl.com/product-detail/xmos/xe216-512-tq128-i20">https://www.e-xfl.com/product-detail/xmos/xe216-512-tq128-i20</a>

## 2 XE216-512-TQ128 Features

### ► Multicore Microcontroller with Advanced Multi-Core RISC Architecture

- 16 real-time logical cores on 2 xCORE tiles
- Cores share up to 1000 MIPS
  - Up to 2000 MIPS in dual issue mode
- Each logical core has:
  - Guaranteed throughput of between  $\frac{1}{5}$  and  $\frac{1}{8}$  of tile MIPS
  - 16x32bit dedicated registers
- 167 high-density 16/32-bit instructions
  - All have single clock-cycle execution (except for divide)
  - 32x32→64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions

### ► USB PHY, fully compliant with USB 2.0 specification

### ► RGMII support, compliant with RGMII v1.3 specification

### ► Programmable I/O

- 81 general-purpose I/O pins, configurable as input or output
  - Up to 25 x 1bit port, 12 x 4bit port, 8 x 8bit port, 4 x 16bit port
  - 4 xCONNECT links
- Port sampling rates of up to 60 MHz with respect to an external clock
- 64 channel endss (32 per tile) for communication with other cores, on or off-chip

### ► Memory

- 512KB internal single-cycle SRAM (max 256KB per tile) for code and data storage
- 16KB internal OTP (max 8KB per tile) for application boot code

### ► Hardware resources

- 12 clock blocks (6 per tile)
- 20 timers (10 per tile)
- 8 locks (4 per tile)

### ► JTAG Module for On-Chip Debug

### ► Security Features

- Programming lock disables debug and prevents read-back of memory contents
- AES bootloader ensures secrecy of IP held on external flash memory

### ► Ambient Temperature Range

- Commercial qualification: 0°C to 70°C
- Industrial qualification: -40°C to 85°C

### ► Speed Grade

- 20: 1000 MIPS

### ► Power Consumption

- 570 mA (typical)

### ► 128-pin TQFP package 0.4 mm pitch

## 4 Signal Description

This section lists the signals and I/O pins available on the XE216-512-TQ128. The device provides a combination of 1bit, 4bit, 8bit and 16bit ports, as well as wider ports that are fully or partially (gray) bonded out. All pins of a port provide either output or input, but signals in different directions cannot be mapped onto the same port.

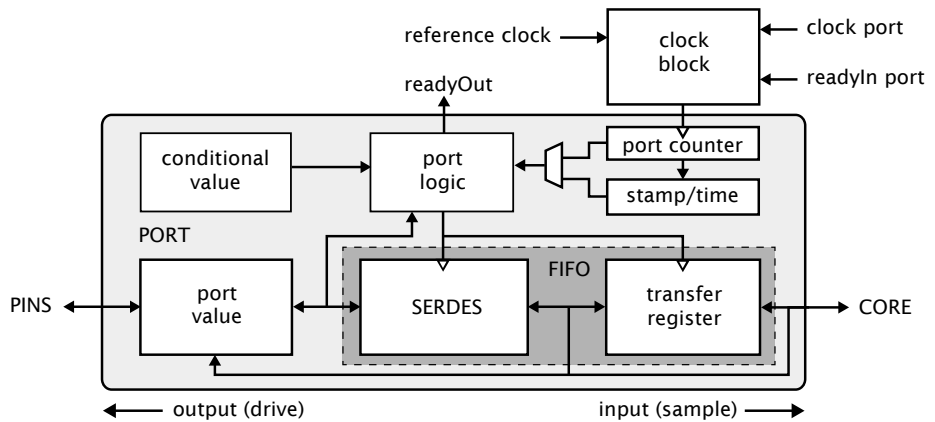
Pins may have one or more of the following properties:

- PD/PU: The IO pin has a weak pull-down or pull-up resistor. The resistor is enabled during and after reset. Enabling a link or port that uses the pin disables the resistor. Thereafter, the resistor can be enabled or disabled under software control. The resistor is designed to ensure defined logic input state for unconnected pins. It should not be used to pull external circuitry. Note that the resistors are highly non-linear and only a maximum pull current is specified in Section 14.2.
- ST: The IO pin has a Schmitt Trigger on its input.
- IOL/IOT/IOR: The IO pin is powered from VDDIOL, VDDIOT, and VDDIOR respectively

Power pins (10)			
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	
USB_VDD	Digital tile power	PWR	
USB_VDD33	USB Analog 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

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.

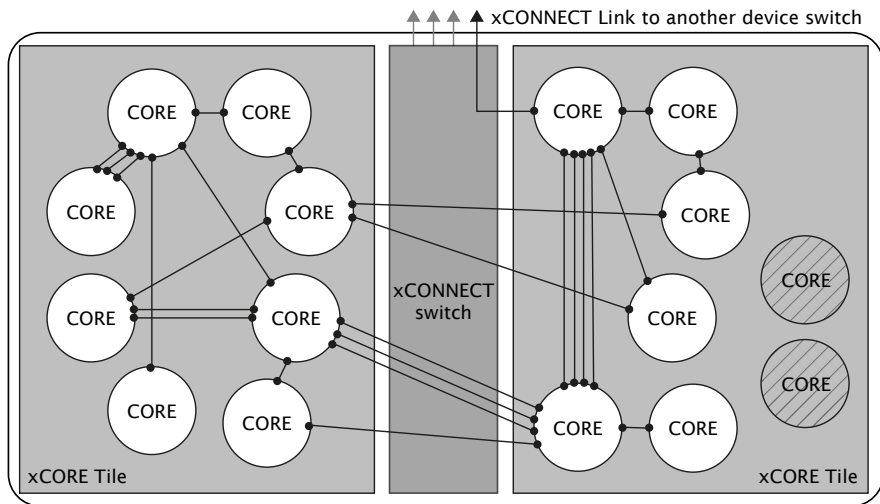
Data is transferred between the pins and core using a FIFO that comprises a SERDES and transfer register, providing options for serialization and buffered data.

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 overrides ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

## 6.4 Clock blocks

xCORE devices include a set of programmable clocks called clock blocks that can be used to govern the rate at which ports execute. Each xCORE tile has six clock blocks: the first clock block provides the tile reference clock and runs at a default frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.



**Figure 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-UE Link Performance and Design Guide, [X2999](#).

## 7 PLL

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock. The initial PLL multiplication value is shown in Figure 7:

**Figure 7:**  
The initial PLL  
multiplier  
values

Oscillator Frequency	Tile Boot Frequency	PLL Ratio	PLL settings		
			OD	F	R
9-25 MHz	144-400 MHz	16	1	63	0

Figure 7 also lists the values of *OD*, *F* and *R*, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

*OD*, *F* and *R* must be chosen so that  $0 \leq R \leq 63$ ,  $0 \leq F \leq 4095$ ,  $0 \leq OD \leq 7$ , and  $260MHz \leq F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \leq 1.3GHz$ . The *OD*, *F*, and *R* values can be modified by writing to the digital node PLL configuration register.

- A 32-bit CRC, or the value 0x0D15AB1E to indicate that no CRC check should be performed.

The program size and CRC are stored least significant byte first. The program is loaded into the lowest memory address of RAM, and the program is started from that address. The CRC is calculated over the byte stream represented by the program size and the program itself. The polynomial used is 0xEDB88320 (IEEE 802.3); the CRC register is initialized with 0xFFFFFFFF and the residue is inverted to produce the CRC.

## 8.1 Boot from QSPI master

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

**Figure 10:**  
QSPI pins

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

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

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

## 8.2 Boot from SPI master

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

**Figure 11:**  
SPI master  
pins

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

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

## 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 256 KB 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 USB PHY

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, XUD, is provided to implement *USB-device* functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 14. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles. Two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.

An external resistor of 43.2 ohm (1% tolerance) should connect USB\_RTUNE to ground, as close as possible to the device.

### 10.1 USB VBUS

USB\_VBUS need not be connected if the device is wholly powered by USB, and the device is used to implement a *USB-device*.

## 10.2 Logical Core Requirements

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

## 11 RGMII

The device has a series of pins that are dedicated to communicate with an RGMII PHY, as per the RGMII v1.3 spec. This can be used to communicate with GBit Ethernet PHYs. The pins and functions are listed in Figure 16. When RGMII mode is enabled (using processor status register 2) these pins can no longer be used as GPIO pins, and will instead be driven directly from an RGMII block that provides DDR to SDR conversion, which in turn is interfaced to a set of ports on Tile 1.

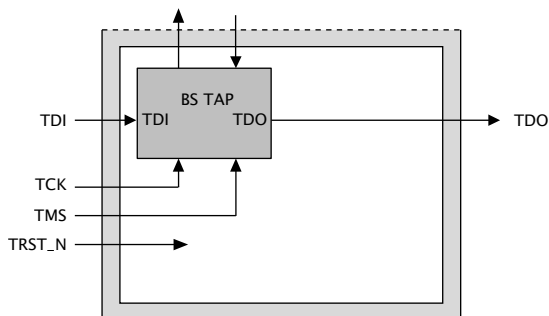
Pin	RGMII Function	
X1D40	TX3	Transmit bit 3
X1D41	TX2	Transmit bit 2
X1D42	TX1	Transmit bit 1
X1D43	TX0	Transmit bit 0
X1D26	TX_CLK	Receive clock (125 MHz)
X1D27	TX_CTL	Transmit data valid/error
X1D28	RX_CLK	Receive clock (125 MHz)
X1D29	RX_CTL	Receive data valid/error
X1D30	RX0	Receive bit 0
X1D31	RX1	Receive bit 1
X1D32	RX2	Receive bit 2
X1D33	RX3	Receive bit 3

**Figure 16:**  
RGMII block  
pin functions

The RGMII block is connected to the ports on Tile 1 as shown in Figure 17. When the RGMII block is enabled, the ports shown can only be used with the RGMII block, and IO pins X1D26..X1D33/X1D40..X1D43 can only be used with the RGMII block. Ports and pins not used in Figure 17 can be used as normal.

The RGMII block generates a clock (configured using processor status register 2), and has the facility to delay the outgoing clock edge, putting it out of phase with the data. The RGMII block translates the double data-rate 4-wire data signals and 1-wire control signal into single-data rate 8-wire TX and DX signals and two control signals. Figure 17 shows how four clock blocks can be used to clock the RGMII ports. One clock block for the TXDATA path, one clock block for the RXDATA path, one clock block to delay the TX\_CLK, and one clock block clocked on a negative valid signal to enable mode switching between 10/100/1000 speeds. Details on how to connect those ports are documented in an application note on RGMII for xCORE-200. The XMOS RGMII software component runs a MAC layer on Tile 1.





**Figure 18:**  
JTAG chain  
structure

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 19.

**Figure 19:**  
IDCODE  
return value

Device Identification Register																																	Bit0				
Version				Part Number																Manufacturer Identity												1					
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1						
0				0								0								6						6						3				3	

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 20. 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 20:**  
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			

## 13 Board Integration

The device has the following power supply pins:

- ▶ VDD pins for the xCORE Tile, including a USB\_VDD pin that powers the USB PHY
- ▶ 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 VDDIOT powers the RGMII IO pins, and must be provided with 2.5V.
- ▶ PLL\_AVDD pins for the PLL
- ▶ OTP\_VCC pins for the OTP

- A USB\_VDD33 pin for the analogue supply to the USB-PHY

Several pins of each type are provided to minimize the effect of inductance within the package, all of which must be connected. The power supplies must be brought up monotonically and input voltages must not exceed specification at any time.

The VDD supply must ramp from 0 V to its final value within 10 ms to ensure correct startup.

The VDDIO and OTP\_VCC supply must ramp to its final value before VDD reaches 0.4 V.

The PLL\_AVDD supply should be separated from the other noisier supplies on the board. The PLL requires a very clean power supply, and a low pass filter (for example, a 4.7  $\Omega$  resistor and 100 nF multi-layer ceramic capacitor) is recommended on this pin.

The following ground pins are provided:

- PLL\_AGND for PLL\_AVDD
- GND for all other supplies

All ground pins must be connected directly to the board ground.

The VDD and VDDIO supplies should be decoupled close to the chip by several 100 nF low inductance multi-layer ceramic capacitors between the supplies and GND (for example, 100nF 0402 for each supply pin). The ground side of the decoupling capacitors should have as short a path back to the GND pins as possible. A bulk decoupling capacitor of at least 10  $\mu$ F 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.

### 13.1 USB connections

USB\_VBUS should be connected to the VBUS pin of the USB connector. A 2.2  $\mu$ F 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.

### 13.2 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

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.1 RAM base address: 0x00

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

<b>0x00:</b> 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.

<b>0x01:</b> 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

### 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>0x04:</b> 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.

<b>0x05:</b> 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](#)

<b>0x06:</b> 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.

---

**0x41:**  
PC of logical  
core 1

---

Bits	Perm	Init	Description
31:0	CRO		Value.

### C.11 PC of logical core 2: 0x42

Value of the PC of logical core 2.

---

**0x42:**  
PC of logical  
core 2

---

Bits	Perm	Init	Description
31:0	CRO		Value.

### C.12 PC of logical core 3: 0x43

Value of the PC of logical core 3.

---

**0x43:**  
PC of logical  
core 3

---

Bits	Perm	Init	Description
31:0	CRO		Value.

### C.13 PC of logical core 4: 0x44

Value of the PC of logical core 4.

---

**0x44:**  
PC of logical  
core 4

---

Bits	Perm	Init	Description
31:0	CRO		Value.

### C.14 PC of logical core 5: 0x45

Value of the PC of logical core 5.

---

**0x45:**  
PC of logical  
core 5

---

Bits	Perm	Init	Description
31:0	CRO		Value.

## F USB PHY Configuration

The USB PHY is connected to the ports shown in section 10.

The *USB PHY* is peripheral 1. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 1, ...)` and `read_periph_32(device, ↪ 1, ...)` for reads and writes).

Number	Perm	Description
0x00	WO	UIFM reset
0x04	RW	UIFM IFM control
0x08	RW	UIFM Device Address
0x0C	RW	UIFM functional control
0x10	RW	UIFM on-the-go control
0x14	RO	UIFM on-the-go flags
0x18	RW	UIFM Serial Control
0x1C	RW	UIFM signal flags
0x20	RW	UIFM Sticky flags
0x24	RW	UIFM port masks
0x28	RW	UIFM SOF value
0x2C	RO	UIFM PID
0x30	RO	UIFM Endpoint
0x34	RW	UIFM Endpoint match
0x38	RW	OTG Flags mask
0x3C	RW	UIFM power signalling
0x40	RW	UIFM PHY control

**Figure 40:**  
Summary

### F.1 UIFM reset: 0x00

A write to this register with any data resets all UIFM state, but does not otherwise affect the phy.

**0x00:**  
UIFM reset

Bits	Perm	Init	Description
31:0	WO		Value.

### F.2 UIFM IFM control: 0x04

General settings of the UIFM IFM state machine.



**0x04:**  
UIFM IFM  
control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7	RW	0	Set to 1 to enable XEVACKMODE mode.
6	RW	0	Set to 1 to enable SOFISTOKEN mode.
5	RW	0	Set to 1 to enable UIFM power signalling mode.
4	RW	0	Set to 1 to enable IF timing mode.
3	RO	-	Reserved
2	RW	0	Set to 1 to enable UIFM linestate decoder.
1	RW	0	Set to 1 to enable UIFM CHECKTOKENS mode.
0	RW	0	Set to 1 to enable UIFM DOTOKENS mode.

### F.3 UIFM Device Address: 0x08

The device address whose packets should be received. 0 until enumeration, it should be set to the assigned value after enumeration.

**0x08:**  
UIFM Device  
Address

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	The enumerated USB device address must be stored here. Only packets to this address are passed on.

### F.4 UIFM functional control: 0x0C

**0x0C:**  
UIFM  
functional  
control

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:2	RW	1	Set to 0 to disable UIFM to UTMI+ OPMODE mode.
1	RW	1	Set to 1 to switch UIFM to UTMI+ TERMSELECT mode.
0	RW	1	Set to 1 to switch UIFM to UTMI+ XCVRSELECT mode.

### F.5 UIFM on-the-go control: 0x10

This register is used to negotiate an on-the-go connection.

0x10: UIFM on-the-go control	Bits	Perm	Init	Description
	31:8	RO	-	Reserved
	7	RW	0	Set to 1 to switch UIFM to EXTVBUSIND mode.
	6	RW	0	Set to 1 to switch UIFM to DRVVBUSEXT mode.
	5	RO	-	Reserved
	4	RW	0	Set to 1 to switch UIFM to UTMI+ CHRGVBUS mode.
	3	RW	0	Set to 1 to switch UIFM to UTMI+ DISCHRGVBUS mode.
	2	RW	0	Set to 1 to switch UIFM to UTMI+ DMPULLDOWN mode.
	1	RW	0	Set to 1 to switch UIFM to UTMI+ DPPULLDOWN mode.
	0	RW	0	Set to 1 to switch UIFM to IDPULLUP mode.

## F.6 UIFM on-the-go flags: 0x14

Status flags used for on-the-go negotiation

0x14: UIFM on-the-go flags	Bits	Perm	Init	Description
	31:6	RO	-	Reserved
	5	RO	0	Value of UTMI+ Bvalid flag.
	4	RO	0	Value of UTMI+ IDGND flag.
	3	RO	0	Value of UTMI+ HOSTDIS flag.
	2	RO	0	Value of UTMI+ VBUSVLD flag.
	1	RO	0	Value of UTMI+ SESSVLD flag.
	0	RO	0	Value of UTMI+ SESEND flag.

### F.7 UIFM Serial Control: 0x18

**0x18:**  
UIFM Serial  
Control

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RO	0	1 if UIFM is in UTMI+ RXRCV mode.
5	RO	0	1 if UIFM is in UTMI+ RXDM mode.
4	RO	0	1 if UIFM is in UTMI+ RXDP mode.
3	RW	0	Set to 1 to switch UIFM to UTMI+ TXSE0 mode.
2	RW	0	Set to 1 to switch UIFM to UTMI+ TXDATA mode.
1	RW	1	Set to 0 to switch UIFM to UTMI+ TXENABLE mode.
0	RW	0	Set to 1 to switch UIFM to UTMI+ FSLSSERIAL mode.

### F.8 UIFM signal flags: 0x1C

Set of flags that monitor line and error states. These flags normally clear on the next packet, but they may be made sticky by using PER\_UIFM\_FLAGS\_STICKY, in which they must be cleared explicitly.

**0x1C:**  
UIFM signal  
flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RW	0	Set to 1 when the UIFM decodes a token successfully (e.g. it passes CRC5, PID check and has matching device address).
5	RW	0	Set to 1 when linestate indicates an SE0 symbol.
4	RW	0	Set to 1 when linestate indicates a K symbol.
3	RW	0	Set to 1 when linestate indicates a J symbol.
2	RW	0	Set to 1 if an incoming datapacket fails the CRC16 check.
1	RW	0	Set to the value of the UTMI_RXACTIVE input signal.
0	RW	0	Set to the value of the UTMI_RXERROR input signal

### F.9 UIFM Sticky flags: 0x20

These bits define the sticky-ness of the bits in the UIFM IFM FLAGS register. A 1 means that bit will be sticky (hold its value until a 1 is written to that bitfield), or normal, in which case signal updates to the UIFM IFM FLAGS bits may be over-written by subsequent changes in those signals.

## I PCB Layout Design Check List

- ✓ This section is a checklist for use by PCB designers using the XS2-UE16A-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 [13.4](#)).
- ☐ 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 RGMII interface

This section can be skipped if you do not have any device connected to the RGMII interface.

- ☐ The RGMII traces are length and impedance matched.

### I.3 Power supply decoupling

- ☐ The decoupling capacitors are all placed close to a supply pin (Section [13](#)).
- ☐ The decoupling capacitors are spaced around the device (Section [13](#)).
- ☐ The ground side of each decoupling capacitor has a direct path back to the center ground of the device.

### I.4 PLL\_AVDD

- ☐ The PLL\_AVDD filter (especially the capacitor) is placed close to the PLL\_AVDD pin (Section [13](#)).

## J Associated Design Documentation

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

## K Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	<a href="#">X7879</a>
XS1 Port I/O Timing	Port timings	<a href="#">X5821</a>
xCONNECT Architecture	Link, switch and system information	<a href="#">X4249</a>
XS1-UE Link Performance and Design Guidelines	Link timings	
XS1-UE Clock Frequency Control	Advanced clock control	