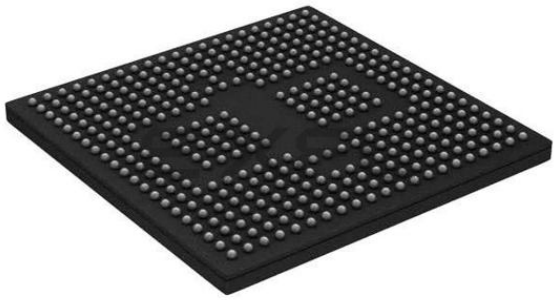

Welcome to E-XFL.COM

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 24-Core
Speed	4000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	176
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	374-LFBGA
Supplier Device Package	374-FBGA (18x18)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xuf224-512-fb374-c40

1 xCORE Multicore Microcontrollers

The xCORE200 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.

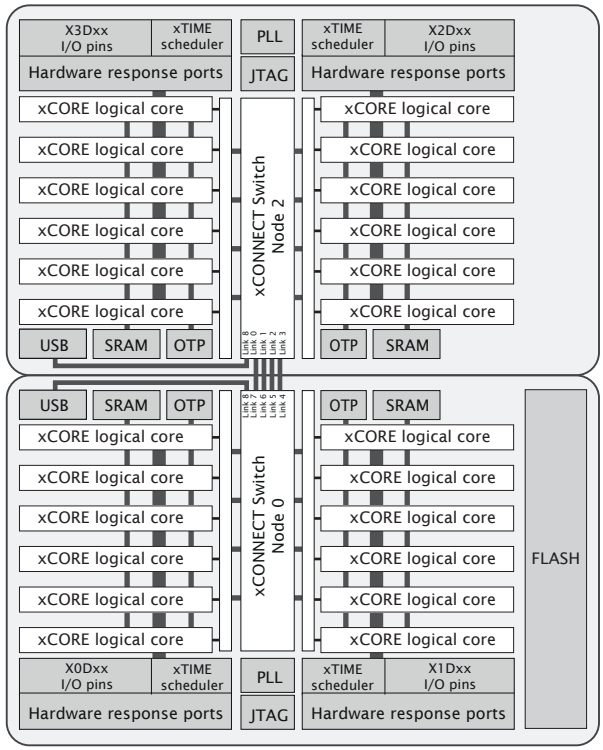

Figure 1: XUF224-512-FB374 block diagram

Key features of the XUF224-512-FB374 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

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	GND	VDDIO	1D X1D11	4E X1D32	4C X1D26	VDDIOT _0	8D X1D41	4C X0D31	4E X0D29	TDI	VDDIO	CLK	TDO	4C X3D32	4C X3D30	VDDIOT _1	4C X2D31	4E X2D29	4E X2D32	VDDIO	GND
B	8D X0D37 A2	1D X0D36 A3	X1D10	4D X1D33	X1D27	8D X1D42	X1D40	4D X0D30	X0D28	4D X2D36	GND	RST_N	TCK	4D X3D33	4D X3D31	X3D27	4D X2D30	X2D28	4E X2D27 A27	X2D26 A26	X2D35 A29
C	1P X0D39 A1	1O X0D38 A4	VDD	4E X1D30	4F X1D28	8D X1D43	GND	4E X0D33	4E X0D32	MODE1	OTP_ VCC	TRST	1C X3D10	4F X3D29	GND	8D X3D43	8D X3D41	4E X2D33	VDD	1J X2D25 A5	1K X2D34 A7
D	1C X0D41 A6	1D X0D40 A24	1D X1D34 A25	1F X1D31	X1D29	GND	VDDIO	NC	GLOBAL _DEBUG	MODE0		TMS	1F X3D11	X3D28	1D X3D26	1D X3D42	1D X3D40	1D X2D70 A14	1H X3D00 A12	1K X3D01 A13	X2D24 A15
E	8D X0D43 A22	8D X0D42 A23	1L X1D35 A20	VDD	VDD	GND	VDDIO	VDD	VDD				VDD	VDD	VDDIO	GND	VDD	VDD	3A X2D69 A18	4A X3D08 A17	4A X3D09 A19
F	1C X1D36 A21	VDDIO	GND	VDD	VDD	VDD	VDD	VDD	VDD	PLL_ AGND	PLL_ AVDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	GND	VDDIO	1C X2D68 A16
G	3A X1D49 A11	3A X1D50 A10	3A X1D51 A9	NC	NC	1O Reserved A8	1M Reserved A12	1K Reserved A13	8D Reserved A14				8D Reserved A15	8D Reserved A16	8D Reserved A17	1P Reserved A18	NC	NC	3A X2D67 A19	3A X2D66 A20	3A X2D65 A21
H	3A X1D53 A4	1M X1D52 A3	VDD																VDD	1M X2D63 A22	3A X2D64 A23
J	4A X1D54 A27	3A X1D55 A26	VDD		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		VDD	3A X2D62 A24	3A X2D61 A25
K	3A X1D58 A5	3A X1D57 A6	3A X1D56 A7		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		3A X2D56 A10	3A X2D57 A9	3A X2D58 A8
L	VDDIO	GND	1K X1D61 A29		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		3A X2D55 A11	GND	VDDIO
M	3A X1D64 A30	3A X1D63 A31	3A X1D62 A32		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		3A X2D54 A12	3A X2D53 A13	3A X2D52 A14
N	1C X1D65 A33	1C X1D66 A34	VDD		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		VDD	1C X2D50 A17	1K X2D51 A16
P	3A X1D68 A35	3A X1D67 A36	VDD																VDD	4B X3D06 A37	4B X3D07 A38
R	3A X1D69 A40	1C X1D70 A41	1K X1D37 A42	NC	NC	NC	NC	8D Reserved A43	8 Reserved A44				1K Reserved A45	1C Reserved A46	NC	NC	NC	NC	3A X2D49 A47	4B X3D04 A48	4B X3D05 A49
T	1O X1D38 A50	VDDIO	GND	VDD	VDD	VDD	USB_ VDD_0	VDD	VDD	VDD	GND	VDD	VDD	VDD	USB_ VDD_1	VDD	VDD	VDD	GND	VDDIO	4A X3D03 A51
U	1C X1D17 A52	1C X1D16 A53	1K X1D39 A54	VDD	VDD	GND	VDDIO	NC	VDD		VDDIO		VDD	VDD	VDDIO	GND	VDD	VDD	NC	4B X2D19 A55	4B X3D02 A56
V	1D X1D19 A57	1D X1D18 A58	1B X0D01 A59	4A X0D02	4A X0D08	1D X0D11	USB_ ID_0	4C X1D14	1J X1D25	1C X0D21	NC	1H X3D23	4B X2D05	1F X2D07	USB_ ID_1	NC	4C X3D15	4C X3D21	1E X2D12	4D X2D17 A60	4D X2D18 A61
W	1C X0D10 A62	1C X1D22 A63	USB_ VDD33_ 0	4A X0D03	4A X0D09	USB_ RTUNE_ 0	GND	4C X1D15	4C X0D14	1E X0D12	1M X0D23	1A X2D00	4B X2D04	4B X2D06	GND	USB_ RTUNE_ 1	4C X3D14	4C X3D20	USB_ VDD33_ 1	1M X2D23	4D X2D16 A64
Y	1D X1D23	1D X0D00	1D X0D04	4B X0D06	1D X1D12	USB_ VBUS_0	X1D24	4D X1D20	4C X0D15	1F X0D13	GND	1D X2D11	1A X2D02	1A X2D08	1F X3D13	VUSB_ BUS_1	4D X2D14	4D X2D20	1F X3D24	X2D13	4D X2D22
AA	GND	VDDIO	4B X0D05	X0D07	1F X1D13	USB_ DM_0	USB_ DP_0	4D X1D21	4C X0D20	1G X0D22	VDDIO	1E X3D12	4A X2D03	4A X2D09	USB_ DM_1	USB_ DP_1	4D X2D15	4C X2D21	1G X3D25	VDDIO	GND

Signal	Function					Type	Properties
X1D64	$X_0L2^1_{in}$				$32A^{13}$	I/O	IO, PD
X1D65	$X_0L2^0_{in}$				$32A^{14}$	I/O	IO, PD
X1D66	$X_0L2^0_{out}$				$32A^{15}$	I/O	IO, PD
X1D67	$X_0L2^1_{out}$				$32A^{16}$	I/O	IO, PD
X1D68	$X_0L2^2_{out}$				$32A^{17}$	I/O	IO, PD
X1D69	$X_0L2^3_{out}$				$32A^{18}$	I/O	IO, PD
X1D70	$X_0L2^4_{out}$				$32A^{19}$	I/O	IO, PD
X2D00		$1A^0$				I/O	IO, PD
X2D02			$4A^0$	$8A^0$	$16A^0$ $32A^{20}$	I/O	IO, PD
X2D03			$4A^1$	$8A^1$	$16A^1$ $32A^{21}$	I/O	IO, PD
X2D04			$4B^0$	$8A^2$	$16A^2$ $32A^{22}$	I/O	IO, PD
X2D05			$4B^1$	$8A^3$	$16A^3$ $32A^{23}$	I/O	IO, PD
X2D06			$4B^2$	$8A^4$	$16A^4$ $32A^{24}$	I/O	IO, PD
X2D07			$4B^3$	$8A^5$	$16A^5$ $32A^{25}$	I/O	IO, PD
X2D08			$4A^2$	$8A^6$	$16A^6$ $32A^{26}$	I/O	IO, PD
X2D09			$4A^3$	$8A^7$	$16A^7$ $32A^{27}$	I/O	IO, PD
X2D11		$1D^0$				I/O	IO, PD
X2D12		$1E^0$				I/O	IO, PD
X2D13		$1F^0$				I/O	IO, PD
X2D14			$4C^0$	$8B^0$	$16A^8$ $32A^{28}$	I/O	IO, PD
X2D15			$4C^1$	$8B^1$	$16A^9$ $32A^{29}$	I/O	IO, PD
X2D16	$X_2L4^4_{in}$		$4D^0$	$8B^2$	$16A^{10}$	I/O	IO, PD
X2D17	$X_2L4^3_{in}$		$4D^1$	$8B^3$	$16A^{11}$	I/O	IO, PD
X2D18	$X_2L4^2_{in}$		$4D^2$	$8B^4$	$16A^{12}$	I/O	IO, PD
X2D19	$X_2L4^1_{in}$		$4D^3$	$8B^5$	$16A^{13}$	I/O	IO, PD
X2D20			$4C^2$	$8B^6$	$16A^{14}$ $32A^{30}$	I/O	IO, PD
X2D21			$4C^3$	$8B^7$	$16A^{15}$ $32A^{31}$	I/O	IO, PD
X2D22		$1G^0$				I/O	IO, PD
X2D23		$1H^0$				I/O	IO, PD
X2D24	$X_2L7^0_{in}$	$1I^0$				I/O	IO, PD
X2D25	$X_2L7^0_{out}$	$1J^0$				I/O	IO, PD
X2D26	$X_2L7^3_{out}$		$4E^0$	$8C^0$	$16B^0$	I/O	IO, PD
X2D27	$X_2L7^4_{out}$		$4E^1$	$8C^1$	$16B^1$	I/O	IO, PD
X2D28			$4F^0$	$8C^2$	$16B^2$	I/O	IO, PD
X2D29			$4F^1$	$8C^3$	$16B^3$	I/O	IO, PD
X2D30			$4F^2$	$8C^4$	$16B^4$	I/O	IO, PD
X2D31			$4F^3$	$8C^5$	$16B^5$	I/O	IO, PD
X2D32			$4E^2$	$8C^6$	$16B^6$	I/O	IO, PD
X2D33			$4E^3$	$8C^7$	$16B^7$	I/O	IO, PD
X2D34	$X_2L7^1_{out}$	$1K^0$				I/O	IO, PD
X2D35	$X_2L7^2_{out}$	$1L^0$				I/O	IO, PD
X2D36		$1M^0$		$8D^0$	$16B^8$	I/O	IO, PD
X2D49	$X_2L5^4_{in}$				$32A^0$	I/O	IO, PD

(continued)

6 Product Overview

The XUF224-512-FB374 is a powerful device that consists of four 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		
20	2000 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 XUF224-512-FB374, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.

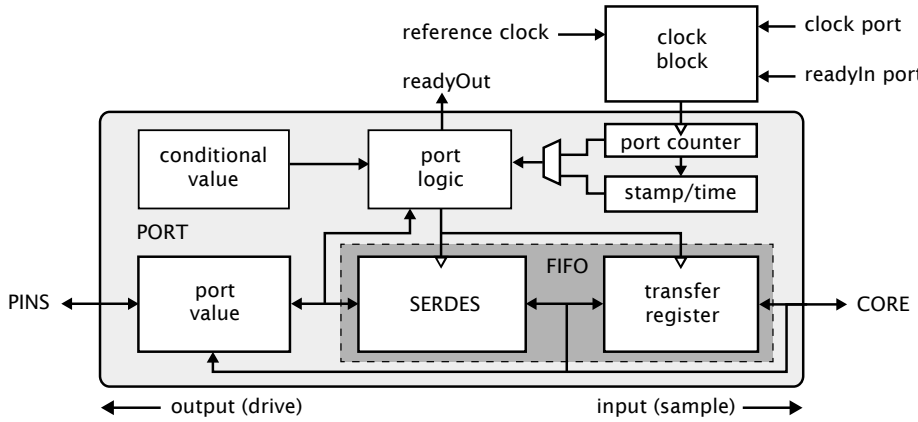

Figure 4: Port block diagram

The port logic can drive its pins high or low, or it can sample the value on its pins, optionally waiting for a particular condition. Ports are accessed using dedicated instructions that are executed in a single processor cycle. xCORE-200 IO pins can be used as *open collector* outputs, where signals are driven low if a zero is output, but left high impedance if a one is output. This option is set on a per-port basis.

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

Each port has a 16-bit counter that can be used to control the time at which data is transferred between the port value and transfer register. The counter values can be obtained at any time to find out when data was obtained, or used to delay I/O until some time in the future. The port counter value is automatically saved as a timestamp, that can be used to provide precise control of response times.

The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

6.4 Clock blocks

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

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.

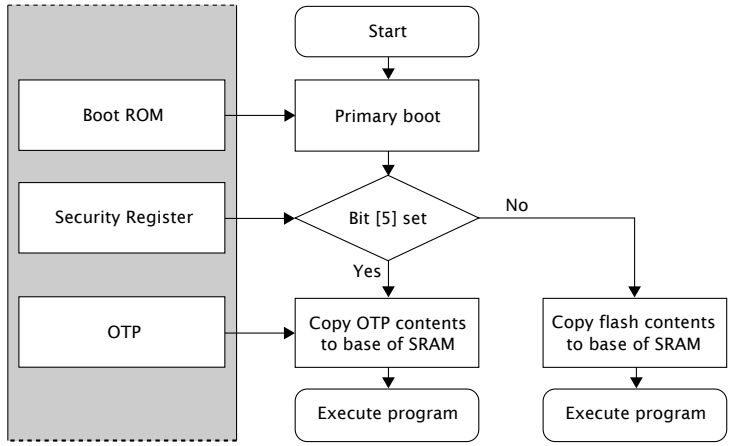

Figure 9: Boot procedure

The boot image has the following format:

- A 32-bit program size s in words.
- Program consisting of $s \times 4$ bytes.
- A 32-bit CRC, or the value 0x0D15AB1E to indicate that no CRC check should be performed.

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

8.1 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 10 provide a strong level of protection and are sufficient for providing strong IP security.

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

Feature	Bit	Description
Disable JTAG	0	The JTAG interface is disabled, making it impossible for the tile state or memory content to be accessed via the JTAG interface.
Disable Link access	1	Other tiles are forbidden access to the processor state via the system switch. Disabling both JTAG and Link access transforms an xCORE Tile into a "secure island" with other tiles free for non-secure user application code.
Secure Boot	5	The xCORE Tile is forced to boot from address 0 of the OTP, allowing the xCORE Tile boot ROM to be bypassed (*see* §8).
Redundant rows	7	Enables redundant rows in OTP.
Sector Lock 0	8	Disable programming of OTP sector 0.
Sector Lock 1	9	Disable programming of OTP sector 1.
Sector Lock 2	10	Disable programming of OTP sector 2.
Sector Lock 3	11	Disable programming of OTP sector 3.
OTP Master Lock	12	Disable OTP programming completely: disables updates to all sectors and security register.
Disable JTAG-OTP	13	Disable all (read & write) access from the JTAG interface to this OTP.
Disable Global Debug	14	Disables access to the DEBUG_N pin.
	21..15	General purpose software accessable security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 10: Security register features

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 128KB 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, libxud_s.a, is provided to implement USB device functionality.

This device has two USB PHYs. One PHY is part of Node 0 and can be connected to either Tile 0 or Tile 1; it uses pins USB_DM, USB_DP, USB_VUBS, USB_ID, and USB_RTUNE. The other PHY is part of Node 2 and can be connected to either Tile 2 or Tile 3; it uses pins USB_2_DM, USB_2_DP, USB_2_VUBS, USB_2_ID, and USB_2_RTUNE. Below we present the configuration of the USB PHY connected to Node 0 (Tiles 0 and 1) an identical configuration is available for Node 2.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 11. 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.

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

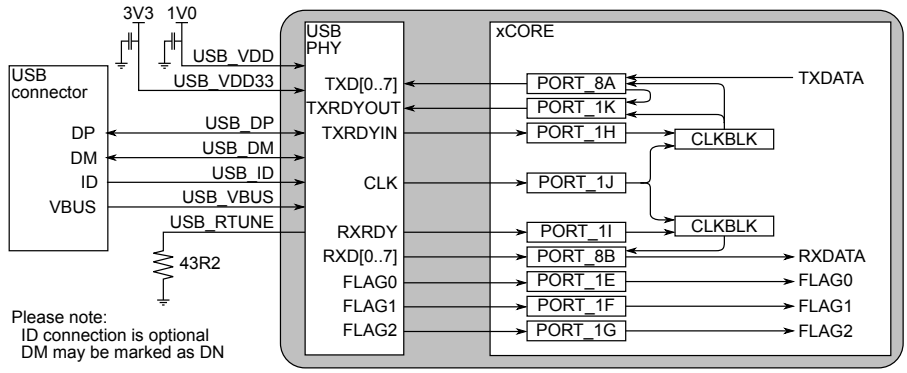

Figure 11: USB port functions

Figure 11 shows how 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.

10.1 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.

13 DC and Switching Characteristics

13.1 Operating Conditions

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDD	Tile DC supply voltage	0.95	1.00	1.05	V	
VDDIO	I/O supply voltage	2.30	3.30	3.60	V	
VDDIOT 3v3	I/O supply voltage	3.135	3.30	3.465	V	
VDDIOT 2v5	I/O supply voltage	2.375	2.50	2.625	V	
USB_VDD	USB tile DC supply voltage	0.95	1.00	1.05	V	
VDD33	Peripheral supply	3.135	3.30	3.465	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
Cl	xCORE Tile I/O load capacitance			25	pF	
Ta	Ambient operating temperature (Commercial)	0		70	°C	
	Ambient operating temperature (Industrial)	-40		85	°C	
Tj	Junction temperature			125	°C	
Tstg	Storage temperature	-65		150	°C	

Figure 17: Operating conditions

13.2 DC Characteristics, VDDIO=3V3

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	A
V(IL)	Input low voltage	-0.30		0.70	V	A
V(OH)	Output high voltage	2.20			V	B, C
V(OL)	Output low voltage			0.40	V	B, C
I(PU)	Internal pull-up current (Vin=0V)	-100			µA	D
I(PD)	Internal pull-down current (Vin=3.3V)			100	µA	D
I(LC)	Input leakage current	-10		10	µA	

Figure 18: DC characteristics

A All pins except power supply pins.

B Pins X1D40, X1D41, X1D42, X1D43, X1D26, X1D27, X3D40, X3D41, X3D42, X3D43, X3D26, and X3D27 are nominal 8 mA drivers, the remainder of the general-purpose I/Os are 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry. In order to pull the pin to the opposite state, a 4K7 resistor is recommended to overome the internal pull current.

13.5 Power Consumption

Figure 22: xCORE Tile currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
I(DDCQ)	Quiescent VDD current		90		mA	A, B, C
PD	Tile power dissipation		325		µW/MIPS	A, D, E, F
IDD	Active VDD current		1140	1400	mA	A, G
I(ADDPLL)	PLL_AVDD current		5	7	mA	H
I(VDD33)	VDD33 current		53.4		mA	I
I(USB_VDD)	USB_VDD current		16.6		mA	J

A Use for budgetary purposes only.

B Assumes typical tile and I/O voltages with no switching activity.

C Includes PLL current.

D Assumes typical tile and I/O voltages with nominal switching activity.

E Assumes 1 MHz = 1 MIPS.

F PD(TYP) value is the usage power consumption under typical operating conditions.

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

H PLL_AVDD = 1.0 V

I HS mode transmitting while driving all 0's data (constant JKJK on DP/DM). Loading of 10 pF. Transfers do not include any interpacket delay.

J HS receive mode; no traffic.

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

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

13.6 Clock

Figure 23: Clock

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f	Frequency	3.25	24	100	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-UF Clock Frequency Control document,

Appendices

A Configuration of the XUF224-512-FB374

The device is configured through banks of registers, as shown in Figure 29.

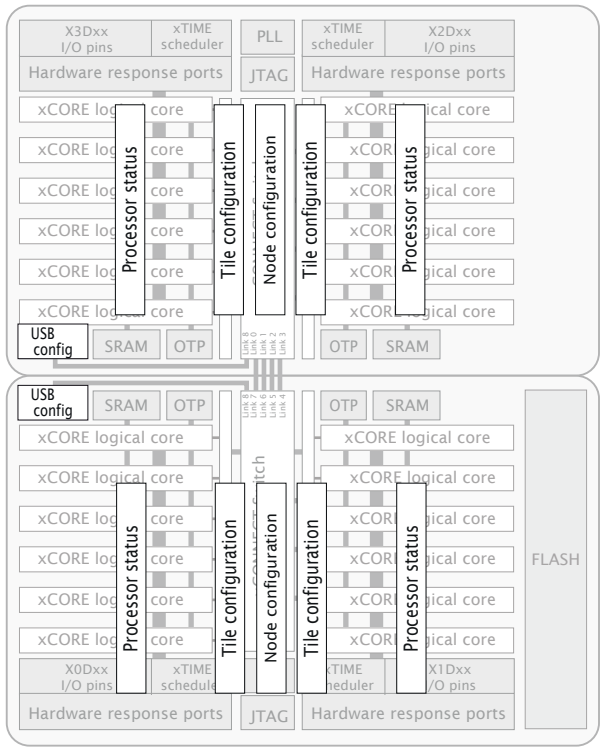

Figure 29: Registers

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

A.1 Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0B. Alternatively, the functions `getps(reg)` and `setps(reg,value)` can be used from XC.

B Processor Status Configuration

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

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

Figure 30: Summary

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

0x02: xCORE Tile control

B.4 xCORE Tile boot status: 0x03

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

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

0x03: xCORE Tile boot status

0x07: Ring Oscillator Value

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

B.8 Ring Oscillator Value: 0x08

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

0x08: Ring Oscillator Value

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

B.9 Ring Oscillator Value: 0x09

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

0x09: Ring Oscillator Value

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

B.10 Ring Oscillator Value: 0x0A

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

0x0A: Ring Oscillator Value

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

B.11 RAM size: 0x0C

The size of the RAM in bytes

C.15 PC of logical core 6: 0x46

Value of the PC of logical core 6.

0x46: PC of logical core 6

Bits	Perm	Init	Description
31:0	CRO		Value.

C.16 PC of logical core 7: 0x47

Value of the PC of logical core 7.

0x47: PC of logical core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

C.17 SR of logical core 0: 0x60

Value of the SR of logical core 0

0x60: SR of logical core 0

Bits	Perm	Init	Description
31:0	CRO		Value.

C.18 SR of logical core 1: 0x61

Value of the SR of logical core 1

0x61: SR of logical core 1

Bits	Perm	Init	Description
31:0	CRO		Value.

C.19 SR of logical core 2: 0x62

Value of the SR of logical core 2

0x62: SR of logical core 2

Bits	Perm	Init	Description
31:0	CRO		Value.

C.20 SR of logical core 3: 0x63

Value of the SR of logical core 3

0x63: SR of logical core 3

Bits	Perm	Init	Description
31:0	CRO		Value.

C.21 SR of logical core 4: 0x64

Value of the SR of logical core 4

0x64: SR of logical core 4

Bits	Perm	Init	Description
31:0	CRO		Value.

C.22 SR of logical core 5: 0x65

Value of the SR of logical core 5

0x65: SR of logical core 5

Bits	Perm	Init	Description
31:0	CRO		Value.

C.23 SR of logical core 6: 0x66

Value of the SR of logical core 6

0x66: SR of logical core 6

Bits	Perm	Init	Description
31:0	CRO		Value.

0x1F: Debug source

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

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 USB Node Configuration

The USB 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).

Figure 33: Summary

Number	Perm	Description
0x00	RO	Device identification register
0x04	RW	Node configuration register
0x05	RW	Node identifier
0x51	RW	System clock frequency
0x80	RW	Link Control and Status

E.1 Device identification register: 0x00

This register contains version information, and information on power-on behavior.

0x00: Device identification register

Bits	Perm	Init	Description
31:24	RO	0x0F	Chip identifier
23:16	RO	-	Reserved
15:8	RO	0x02	Revision number of the USB block
7:0	RO	0x00	Version number of the USB block

E.2 Node configuration register: 0x04

This register is used to set the communication model to use (1 or 3 byte headers), and to prevent any further updates.

0x04: Node configuration register

Bits	Perm	Init	Description
31	RW	0	Set to 1 to disable further updates to the node configuration and link control and status registers.
30:1	RO	-	Reserved
0	RW	0	Header mode. 0: 3-byte headers; 1: 1-byte headers.

0x2C: UIFM PID

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

F.13 UIFM Endpoint: 0x30

The last endpoint seen

0x30: UIFM Endpoint

Bits	Perm	Init	Description
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.

0x34: UIFM Endpoint match

Bits	Perm	Init	Description
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

0x38: OTG Flags mask

Bits	Perm	Init	Description
31:0	RW	0	Data

F.16 UIFM power signalling: 0x3C

0x3C: UIFM power signalling

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

L Revision History

Date	Description
2015-03-20	Preliminary release
2015-04-14	Added RST to pins to be pulled hard, and removed reference to TCK from Errata Removed TRST_N references in packages that have no TRST_N New diagram for boot from embedded flash showing ports Pull up requirements for shared clock and external resistor for QSPI
2015-05-06	Removed references to DEBUG_N
2015-07-09	Updated electrical characteristics - Section 13
2015-08-19	Added I(USB_VDD) - Section 13 Added USB layout guidelines - Section 12
2015-08-27	Updated part marking and product code - Section 15
2015-11-23	Updated status of X2D04, X2D05, X2D06, X2D07 during boot - Section 8 Updated Schematics Design Checklist: GPIO for X2D04, X2D05, X2D06, X2D07 during boot - Section H
2015-12-18	Clarified connectivity of internal and external xCONNECT links - Sections 3 and 4 Made pin names canonical - Sections 3 and 4 Updated JTAG diagram - Section 11 Removed references to 400MHz parts - Section 13
2016-01-05	Updated signal tables to use VDDIO - Section 4 Updated IDD value - Section 13 Updated land pattern description - Section 12.3
2016-04-20	Typical internal pull-up and pull down current diagrams added - Section 13