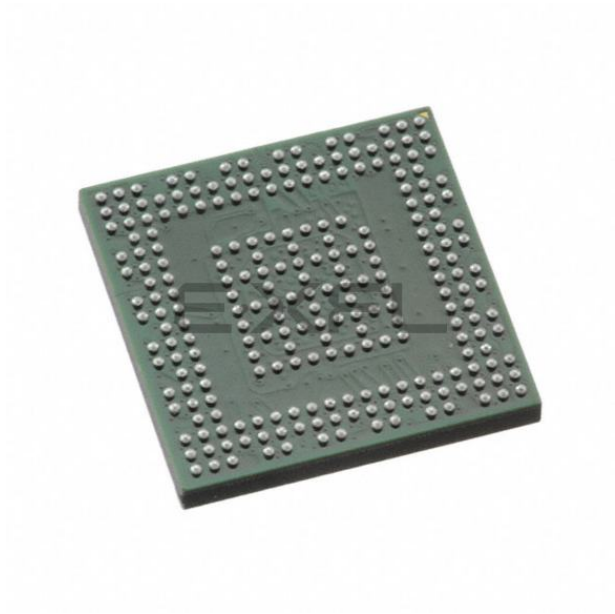

Welcome to E-XFL.COM

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 12-Core
Speed	2000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	104
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	236-LFBGA
Supplier Device Package	236-FBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xuf212-512-fb236-c20

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	GND	VDDIOL	VDDIOL		TCK	CLK		4F X1D31	4F X1D29		8D X1D41	OTP_VCC		NC	MODE[0]		4F X0D29	VDDIOR	GND
B	1M X0D36	VDDIOL	VDDIOL	TDO	TMS	TRST_N	4E X1D33	4E X1D32	4F X1D28	4E X1D26	8D X1D42	OTP_VCC	NC	NC	MODE[1]	4E X0D33	4E X0D32	VDDIOR	VDDIOR
C	1N X0D37	1O X0D38	VDDIOL	TDI	DEBUG_N	RST_N	1C X1D10	1D X1D11	4F X1D30	4E X1D27	8D X1D43	8D X1D40	NC	NC	4F X0D31	4F X0D30	4F X0D28	4E X0D26	4E X0D27
D		1P X0D39	8D X0D40														1K X0D34	1L X0D35	
E	8D X0D43	8D X0D42	8D X0D41														1J X0D25	1I X0D24	1S X1D01
F	1K X1D34	1L X1D35	1M X1D36				NC	VDD	VDD	VDDIOT	VDD	VDD	PLL_AVDD	PLL_AGND			4A X1D08	4A X1D09	1A X1D00
G		32A X1D49	32A X1D50			VDD		GND		GND		GND		VDD			32A X0D69	32A X0D70	
H	32A X1D53	32A X1D52	32A X1D51			VDD	GND	GND	GND	GND	GND	GND	GND	VDD			32A X0D68	32A X0D67	32A X0D66
J	32A X1D54	32A X1D55	32A X1D56			VDD		GND		GND		GND		VDD			32A X0D63	32A X0D64	32A X0D65
K		32A X1D58	32A X1D57			VDD	GND	GND	GND	GND	GND	GND	GND	VDD			32A X0D62	32A X0D61	
L	32A X1D63	32A X1D62	32A X1D61			VDD		GND		GND		GND		VDD			32A X0D58	32A X0D57	32A X0D56
M	32A X1D64	32A X1D65	32A X1D66			VDD	GND	GND	GND	GND	GND	GND	GND	VDD			32A X0D53	32A X0D54	32A X0D55
N		32A X1D67	32A X1D68			VDD		GND		GND		GND		VDD			32A X0D51	32A X0D52	
P	32A X1D70	32A X1D69	1N X1D37			VDD	VDD	VDD	USB_VDD	USB_VDD	VDD	VDD	VDD	NC			4B X1D07	32A X0D50	32A X0D49
R	1O X1D38	1P X1D39	4D X1D17														4A X1D03	4B X1D05	4B X1D06
T		4D X1D16	4D X1D18														4A X1D02	4B X1D04	
U	1C X0D10	1B X0D01	4D X1D19	1A X0D00	1D X0D11	4B X0D07	1E X1D12	USB_VDD33	USB_VBUS	USB_ID	USB_VSSAC	NC	1I X1D24	1G X0D22	1F X0D13	1H X0D23	4D X0D19	4D X0D18	4D X0D17
V	1G X1D22	VDDIOL	VDDIOL	4B X0D04	4B X0D06	4A X0D03	4A X0D08	4A X0D09	USB_DM	USB_DP	4C X1D21	4C X1D14	1J X1D25	4C X0D21	4C X0D14	1E X0D12	VDDIOR	VDDIOR	4D X0D16
W	GND	VDDIOL	1H X1D23		4B X0D05	4A X0D02		1F X1D13	USB_RTUNE		4C X1D20	4C X1D15		4C X0D20	4C X0D15		VDDIOR	VDDIOR	GND

Signal	Function						Type	Properties
X1D20			$4C^{2}$	$8B^{6}$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X1D21			$4C^{3}$	$8B^{7}$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X1D22	$X_0L3^4_{out}$	$1G^{0}$					I/O	IOL, PD
X1D23		$1H^{0}$					I/O	IOL, PD
X1D24		$1I^{0}$					I/O	IOR, PD
X1D25		$1J^{0}$					I/O	IOR, PD
X1D26			$4E^{0}$	$8C^{0}$	$16B^{0}$		I/O	IOT, PD
X1D27			$4E^{1}$	$8C^{1}$	$16B^{1}$		I/O	IOT, PD
X1D28			$4F^{0}$	$8C^{2}$	$16B^{2}$		I/O	IOT, PD
X1D29			$4F^{1}$	$8C^{3}$	$16B^{3}$		I/O	IOT, PD
X1D30			$4F^{2}$	$8C^{4}$	$16B^{4}$		I/O	IOT, PD
X1D31			$4F^{3}$	$8C^{5}$	$16B^{5}$		I/O	IOT, PD
X1D32			$4E^{2}$	$8C^{6}$	$16B^{6}$		I/O	IOT, PD
X1D33			$4E^{3}$	$8C^{7}$	$16B^{7}$		I/O	IOT, PD
X1D34	$X_0L0^2_{out}$	$1K^{0}$					I/O	IOL, PD
X1D35	$X_0L0^3_{out}$	$1L^{0}$					I/O	IOL, PD
X1D36	$X_0L0^4_{out}$	$1M^{0}$		$8D^{0}$	$16B^{8}$		I/O	IOL, PD
X1D37	$X_0L3^4_{in}$	$1N^{0}$		$8D^{1}$	$16B^{9}$		I/O	IOL, PD
X1D38	$X_0L3^3_{in}$	$1O^{0}$		$8D^{2}$	$16B^{10}$		I/O	IOL, PD
X1D39	$X_0L3^2_{in}$	$1P^{0}$		$8D^{3}$	$16B^{11}$		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
X1D49	$X_0L1^4_{in}$					$32A^{0}$	I/O	IOL, PD
X1D50	$X_0L1^3_{in}$					$32A^{1}$	I/O	IOL, PD
X1D51	$X_0L1^2_{in}$					$32A^{2}$	I/O	IOL, PD
X1D52	$X_0L1^1_{in}$					$32A^{3}$	I/O	IOL, PD
X1D53	$X_0L1^0_{in}$					$32A^{4}$	I/O	IOL, PD
X1D54	$X_0L1^0_{out}$					$32A^{5}$	I/O	IOL, PD
X1D55	$X_0L1^1_{out}$					$32A^{6}$	I/O	IOL, PD
X1D56	$X_0L1^2_{out}$					$32A^{7}$	I/O	IOL, PD
X1D57	$X_0L1^3_{out}$					$32A^{8}$	I/O	IOL, PD
X1D58	$X_0L1^4_{out}$					$32A^{9}$	I/O	IOL, PD
X1D61	$X_0L2^4_{in}$					$32A^{10}$	I/O	IOL, PD
X1D62	$X_0L2^3_{in}$					$32A^{11}$	I/O	IOL, PD
X1D63	$X_0L2^2_{in}$					$32A^{12}$	I/O	IOL, PD
X1D64	$X_0L2^1_{in}$					$32A^{13}$	I/O	IOL, PD
X1D65	$X_0L2^0_{in}$					$32A^{14}$	I/O	IOL, PD
X1D66	$X_0L2^0_{out}$					$32A^{15}$	I/O	IOL, PD
X1D67	$X_0L2^1_{out}$					$32A^{16}$	I/O	IOL, PD
X1D68	$X_0L2^2_{out}$					$32A^{17}$	I/O	IOL, PD
X1D69	$X_0L2^3_{out}$					$32A^{18}$	I/O	IOL, PD

(continued)

5 Example Application Diagram

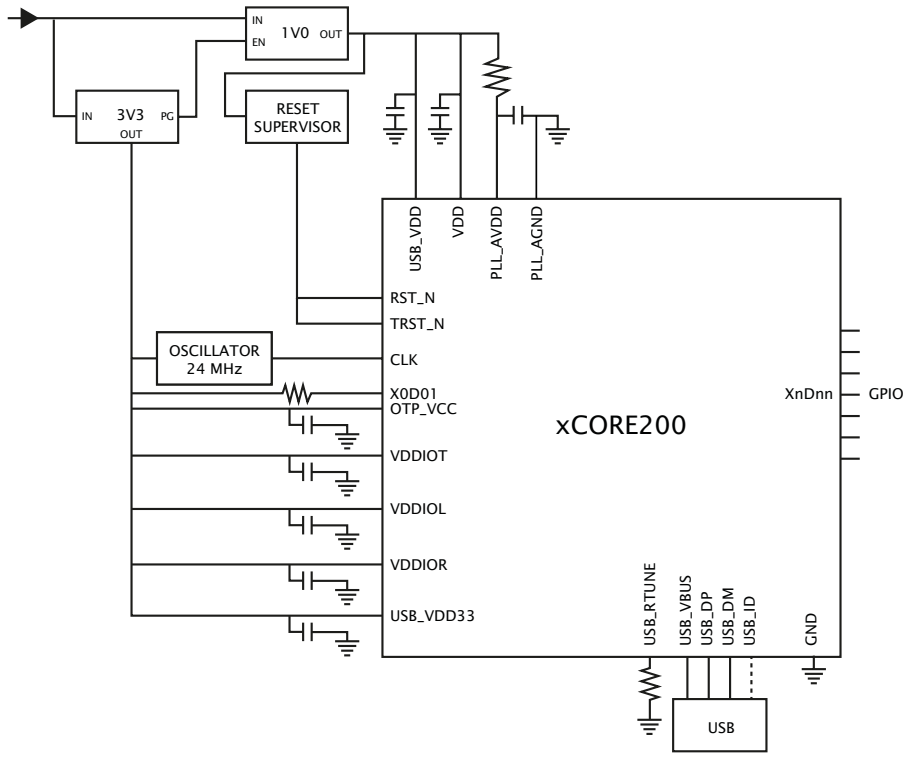

Figure 2: Simplified Reference Schematic

- see Section 10 for details on the USB PHY
- see Section 12 for details on the power supplies and PCB design

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

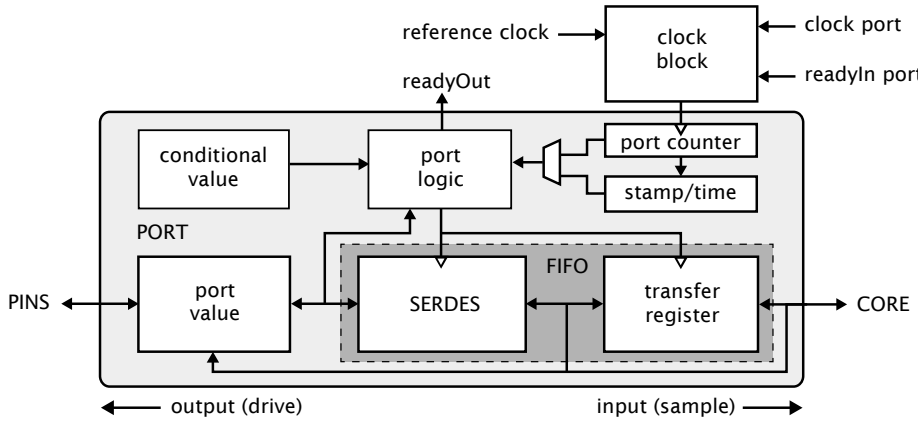

Figure 4: Port block diagram

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.

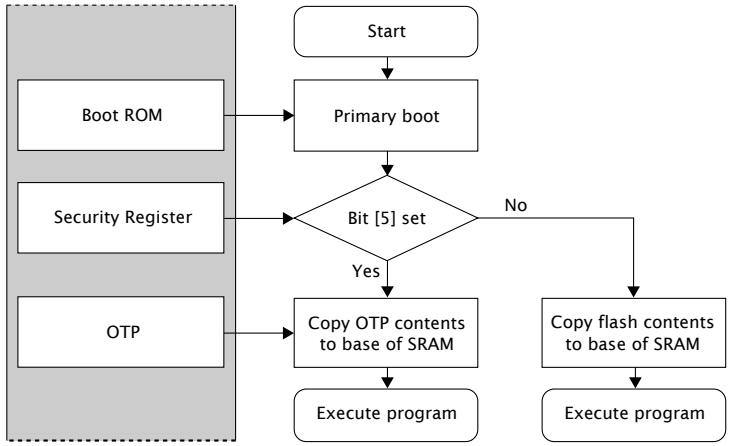

Figure 9: Boot procedure

- 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 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.

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

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 256KB 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),

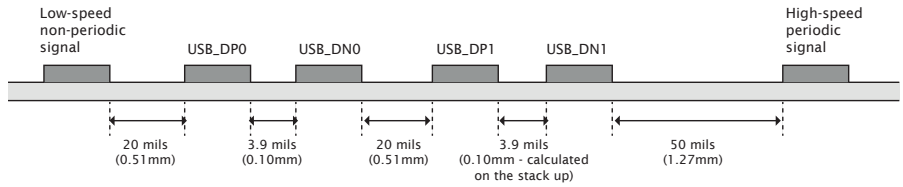

Figure 16: USB trace separation showing a low speed signal, two differential pairs and a high-speed clock

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.

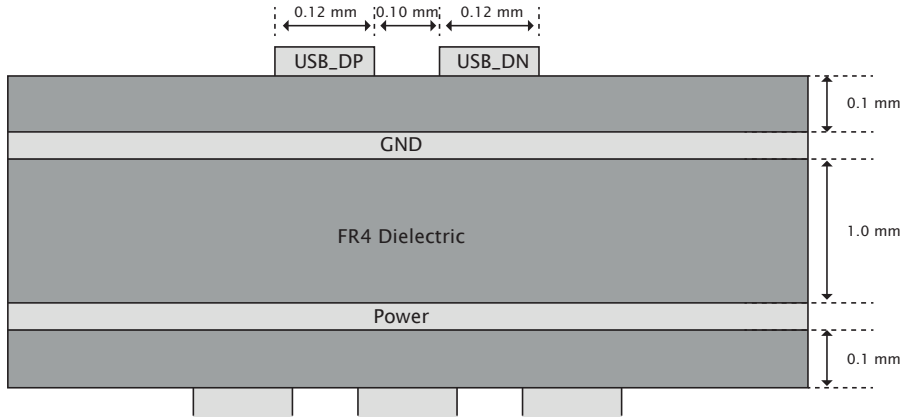

Figure 17: Example USB board stack

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS2-UF12A-512-FB236 are on the top layer) closest to GND. Reference planes should be below the transmission lines in order to maintain control of the trace impedance.

We recommend that the high-speed clock and high-speed USB differential pairs are routed first before any other routing. When routing high speed USB signals, the following guidelines should be followed:

- High speed differential pairs should be routed together.
- High-speed USB signal pair traces should be trace-length matched. Maximum trace-length mismatch should be no greater than 4mm.
- Ensure that high speed signals (clocks, USB differential pairs) are routed as far away from off-board connectors as possible.

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.

13.7 xCORE Tile 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	

Figure 25: I/O AC characteristics

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.

13.8 xConnect 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.

Figure 26: Link performance

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.

13.9 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.

Figure 27: JTAG timing

All JTAG operations are synchronous to TCK apart from the global asynchronous reset TRST_N.

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.5 Security configuration: 0x05

Copy of the security register as read from OTP.

0x05: Security configuration

Bits	Perm	Init	Description
31	RW		Disables write permission on this register
30:15	RO	-	Reserved
14	RW		Disable access to XCore's global debug
13	RO	-	Reserved
12	RW		lock all OTP sectors
11:8	RW		lock bit for each OTP sector
7	RW		Enable OTP reduanacy
6	RO	-	Reserved
5	RW		Override boot mode and read boot image from OTP
4	RW		Disable JTAG access to the PLL/BOOT configuration registers
3:1	RO	-	Reserved
0	RW		Disable access to XCore's JTAG debug TAP

B.6 Ring Oscillator Control: 0x06

There are four free-running oscillators that clock four counters. The oscillators can be started and stopped using this register. The counters should only be read when the ring oscillator has been stopped for at least 10 core clock cycles (this can be achieved by inserting two nop instructions between the SETPS and GETPS). The counter values can be read using four subsequent registers. The ring oscillators are asynchronous to the xCORE tile clock and can be used as a source of random bits.

0x06: Ring Oscillator Control

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Core ring oscillator enable.
0	RW	0	Peripheral ring oscillator enable.

B.7 Ring Oscillator Value: 0x07

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

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 countains 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.

0x30 .. 0x33: Instruction breakpoint address

Bits	Perm	Init	Description
31:0	DRW		Value.

B.22 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 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 PC == IBREAK_ADDR. When 1 = break when PC != IBREAK_ADDR.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.23 Data watchpoint address 1: 0x50 .. 0x53

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

0x50 .. 0x53: Data watchpoint address 1

Bits	Perm	Init	Description
31:0	DRW		Value.

B.24 Data watchpoint address 2: 0x60 .. 0x63

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

0x60 .. 0x63: Data watchpoint address 2

Bits	Perm	Init	Description
31:0	DRW		Value.

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 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.

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

0x0D: Directions 8-15

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose dimension is F.
27:24	RW	0	The direction for packets whose dimension is E.
23:20	RW	0	The direction for packets whose dimension is D.
19:16	RW	0	The direction for packets whose dimension is C.
15:12	RW	0	The direction for packets whose dimension is B.
11:8	RW	0	The direction for packets whose dimension is A.
7:4	RW	0	The direction for packets whose dimension is 9.
3:0	RW	0	The direction for packets whose dimension is 8.

D.12 DEBUG_N configuration, tile 0: 0x10

Configures the behavior of the DEBUG_N pin.

0x10: DEBUG_N configuration, tile 0

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set 1 to enable GlobalDebug to generate debug request to XCore.
0	RW	0	Set 1 to enable inDebug bit to drive GlobalDebug.

D.13 DEBUG_N configuration, tile 1: 0x11

Configures the behavior of the DEBUG_N pin.

0x11: DEBUG_N configuration, tile 1

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set 1 to enable GlobalDebug to generate debug request to XCore.
0	RW	0	Set 1 to enable inDebug bit to drive GlobalDebug.

D.14 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, 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.3 Node identifier: 0x05

0x05: Node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	16-bit node identifier. This does not need to be set, and is present for compatibility with XS1-switches.

E.4 System clock frequency: 0x51

0x51: System clock frequency

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	25	Oscillator clock frequency in MHz rounded up to the nearest integer value. Only values between 5 and 100 MHz are valid - writes outside this range are ignored and will be NACKed. This field must be set on start up of the device and any time that the input oscillator clock frequency is changed. It must contain the system clock frequency in MHz rounded up to the nearest integer value.

E.5 Link Control and Status: 0x80

0x80: Link Control and Status

Bits	Perm	Init	Description
31:28	RO	-	Reserved
27	RO		Rx buffer overflow or illegal token encoding received.
26	RO	0	This end of the xlink has issued credit to allow the remote end to transmit
25	RO	0	This end of the xlink has credit to allow it to transmit.
24	WO		Clear this end of the xlink's credit and issue a HELLO token.
23	WO		Reset the receiver. The next symbol that is detected will be the first symbol in a token.
22	RO	-	Reserved
21:11	RW	1	Specify min. number of idle system clocks between two continuous symbols witin a transmit token -1.
10:0	RW	1	Specify min. number of idle system clocks between two continuous transmit tokens -1.

I PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS2-UF12A-512-FB236. Each of the following sections contains items to check for each design.

I.1 Ground Plane

☐ Each ground ball has a via to 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).

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 - Section 15
2016-04-20	Typical internal pull-up and pull down current diagrams added - Section 13
2017-02-02	Updated USB VBUS wiring description with bus-powered usb-device instructions - Section 10

XMOS®