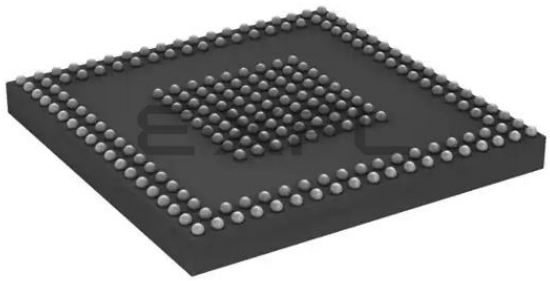

Welcome to E-XFL.COM

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	Active
Core Processor	XCore
Core Size	32-Bit 10-Core
Speed	1000MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	73
Program Memory Size	128KB (32K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	217-LFBGA
Supplier Device Package	217-FBGA (16x16)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xs1-u10a-128-fb217-i10

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	8B X1D05	8B X1D06	8B X1D07	8A X1D08	8A X1D09	8C X1D10	8C X1D11	8C X1D12	8C X1D13	8C X1D14	8C X1D15	8C X1D16	8C X1D17	8C X1D18	8C X1D19	8C X1D20	8C X1D21	8C X1D22	8C X1D23
B	8A X1D04	8A X1D53	8A X1D54	8A X1D55	8A X1D56	8A X1D57	8A X1D58	8A X1D61	8A X1D62	8A X1D63	8A X1D64	8A X1D65	8A X1D66	8A X1D67	8A X1D68	8A X1D69	8A X1D70	8A X1D24	8A X1D25
C	4A X1D03	32A X1D52																4E X1D26	4E X1D27
D	4A X1D02	32A X1D51																4E X1D33	4E X1D32
E	8B X1D01	32A X1D50																1L X1D35	1K X1D34
F	1 X1D00	32A X1D49				GND	GND	GND	GND	GND	GND	GND	GND	GND				VDDIO_ OUT	1J X1D36
G	USB_ DN	USB VBUS				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[4]	1H X1D37
H	USB_ DP	USB_ ID				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[3]	1G X1D38
J	X0D43/ WAKE	RST_N				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[2]	1F X1D39
K	VDDIO	VDDIO				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[1]	TDO
L	ADC6	ADC7				OSC_ EXT_N	GND	GND	GND	GND	GND	GND	GND	GND				MODE[0]	TCK
M	ADC4	ADC5				NC	NC	GND	GND	GND	GND	GND	GND	GND				DEBUG_ N	TMS
N	AVDD	AVSS				NC	NC	GND	GND	GND	GND	GND	GND	GND				NC	TDI
P	ADC2	ADC3				AVSS	GND	GND	GND	GND	GND	GND	GND	GND				NC	1L X0D35
R	ADC0	ADC1																NC	1A X0D00
T	NC	NC																NC	1B X0D01
U	XI/ CLK	NC																NC	1D X0D10
V	XO	NC	VDDCORE	PGND	PGND	SW1	VSUP	VDD1V8	PGND	PGND	SW2	NC	1I X0D24	4C X0D21	4D X0D19	4D X0D17	4C X0D15	NC	1D X0D11
W	VSUP	NC	VDDCORE	VDDCORE	PGND	SW1	VSUP	VDD1V8	VDD1V8	PGND	SW2	NC	1G X0D22	4C X0D20	4D X0D18	4D X0D16	4C X0D14	1F X0D13	1E X0D12

Signal	Function		Type	Properties
X1D51	XLC^{2}_{out}	$32A^{2}$	I/O	PD_S
X1D52	XLC^{1}_{out}	$32A^{3}$	I/O	PD_S
X1D53	XLC^{0}_{out}	$32A^{4}$	I/O	PD_S
X1D54	XLC^{0}_{in}	$32A^{5}$	I/O	PD_S
X1D55	XLC^{1}_{in}	$32A^{6}$	I/O	PD_S
X1D56	XLC^{2}_{in}	$32A^{7}$	I/O	PD_S
X1D57	XLC^{3}_{in}	$32A^{8}$	I/O	PD_S
X1D58	XLC^{4}_{in}	$32A^{9}$	I/O	PD_S
X1D61	XLD^{4}_{out}	$32A^{10}$	I/O	PD_S
X1D62	XLD^{3}_{out}	$32A^{11}$	I/O	PD_S
X1D63	XLD^{2}_{out}	$32A^{12}$	I/O	PD_S
X1D64	XLD^{1}_{out}	$32A^{13}$	I/O	PD_S
X1D65	XLD^{0}_{out}	$32A^{14}$	I/O	PD_S
X1D66	XLD^{0}_{in}	$32A^{15}$	I/O	PD_S
X1D67	XLD^{1}_{in}	$32A^{16}$	I/O	PD_S
X1D68	XLD^{2}_{in}	$32A^{17}$	I/O	PD_S
X1D69	XLD^{3}_{in}	$32A^{18}$	I/O	PD_S
X1D70	XLD^{4}_{in}	$32A^{19}$	I/O	PD_S

If a different tile frequency is required (eg, 500 MHz), then the PLL must be reprogrammed after boot to provide the required tile frequency. The XMOS tools perform this operation by default. Further details on configuring the clock can be found in the XS1-L Clock Frequency Control document, X1433.

9 Boot Procedure

The device is kept in reset by driving RST_N low. When in reset, all GPIO pins are high impedance. When the device is taken out of reset by releasing RST_N the processor starts its internal reset process. After approximately 750,000 input clocks, all GPIO pins have their internal pull-resistor enabled, and the processor boots at a clock speed that depends on MODE0 and MODE1.

The processor boot procedure is illustrated in Figure 9. In normal usage, MODE[4:2] controls the boot source according to the table in Figure 10. If bit 5 of the security register (*see* §10.1) is set, the device boots from OTP.

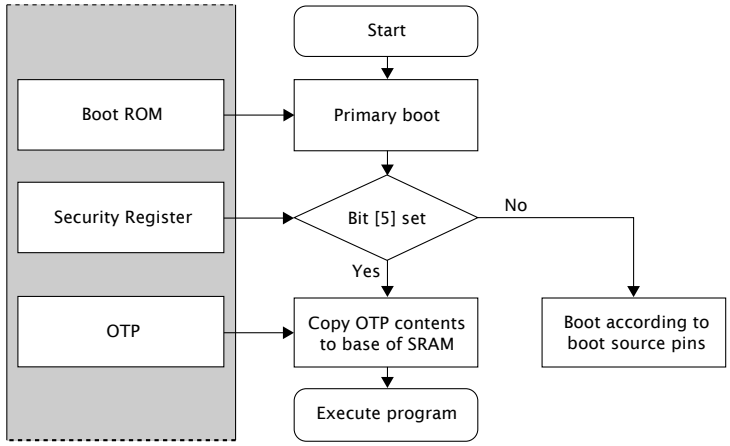

Figure 9: Boot procedure

MODE [4]	MODE [3]	MODE [2]	Boot Source
X	0	0	None: Device waits to be booted via JTAG
X	0	1	Reserved
0	1	0	Tile0 boots from link B, Tile1 from channel end 0 via Tile0
0	1	1	Tile0 boots from SPI, Tile1 from channel end 0 via Tile0
1	1	0	Tile0 and Tile1 independently enable link B and internal links (E, F, G, H), and boot from channel end 0
1	1	1	Tile0 and Tile 1 boot from SPI independently

Figure 10: Boot source pins

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.

9.1 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.

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

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

9.2 Boot from xConnect Link

If set to boot from an xConnect Link, the processor enables Link B around 200 ns after the boot process starts. Enabling the Link switches off the pull-down on resistors X0D16..X0D19, drives X0D16 and X0D17 low (the initial state for the Link), and monitors pins X0D18 and X0D19 for boot-traffic. X0D18 and X0D19 must be low at this stage. If the internal pull-down is too weak to drain any residual charge, external pull-downs of 10K may be required on those pins.

The boot-rom on the core will then:

1. Allocate channel-end 0.

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 18. The OTP User ID field is read from bits [22:31] of the security register on xCORE Tile 0, *see* §10.1 (all zero on unprogrammed devices).

Figure 18: USERCODE return value

Bit31										Usercode Register																				Bit0
OTP User ID										Unused				Silicon Revision																
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				2				C				0				0				0			

16 Board Integration

XS1-U10A-128-FB217 devices are optimized for layout on low cost PCBs using standard design rules. Careful layout is required to maximize the device performance. XMOS therefore recommends that the guidelines in this section are followed when laying out boards using the device.

The XS1-U10A-128-FB217 includes two DC-DC buck converters that take input voltages between 3.3-5V and output the 1.8V and 1.0V circuits required by the digital core and analogue peripherals. The DC-DC converters should have a 4.7uF X5R or X7R ceramic capacitor and a 100nF X5R or X7R ceramic capacitor on the VSUP input pins V7 and W7. These capacitors must be placed as close as possible to the those pins (within a maximum of 5mm), with the routing optimized to minimize the inductance and resistance of the traces.

The SW output pin must have an LC filter on the output with a 4.7uH inductor and 22uF X5R capacitor. The capacitor must have maximum ESR value of 0.015R, and the inductor should have a maximum DCR value of 0.07R. A list of suggested inductors is in Figure 19.

Figure 19: Example 4.7 μH inductors

	Part number	Current	Max DCR	Package
Wurth	744043004	1550 mA	70 $m\Omega$	4.8 x 4.8 mm
Murata	LQH55DN4R7M03L	2700 mA	57 $m\Omega$	5750 (2220)

The traces from the SW output pins to the inductor and from the output capacitor back to the VDD pins must be routed to minimize the coupling between them.

The power supplies must be brought up monotonically and input voltages must not exceed specification at any time.

The VDDIO supply to the XS1-U10A-128-FB217 requires a 100nF X5R or X7R ceramic decoupling capacitor placed as close as possible to the supply pins. VDDIO_OUT is the switched IO supply; it is only supplied when the chip is AWAKE. This pin can be used to provide extra decoupling, or it can be used to switch other devices off during sleep mode, for example a SPI flash. No more than 240 mA should be drawn on VDDIO at any time: this includes any supply sourced statically (e.g., driving a LED from a GPIO pin), any dynamic power consumption (e.g., toggling a GPIO pin at a high frequency) and any devices powered through VDDIO_OUT.

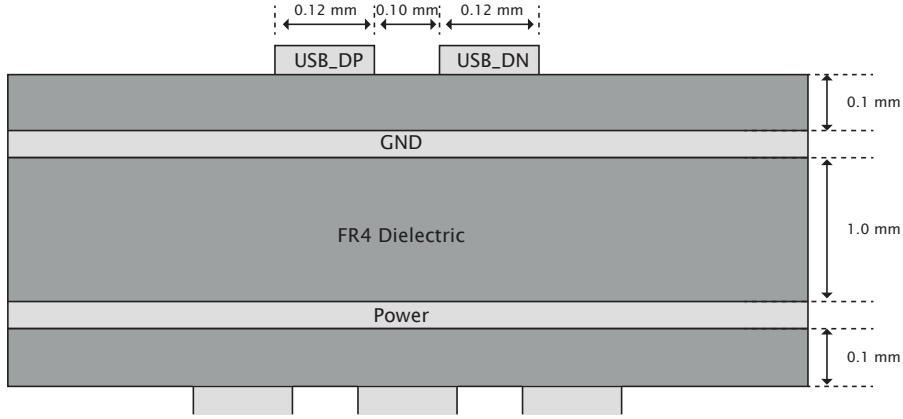

Figure 21: Example USB board stack

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS1-U10A-128-FB217 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.
- High-speed clock and periodic signal traces that run parallel should be at least 1.27mm away from USB_DP/USB_DN (see Figure 20).
- Low-speed and non-periodic signal traces that run parallel should be at least 0.5mm away from USB_DP/USB_DN (see Figure 20).
- Route high speed USB signals on the top of the PCB wherever possible.
- Route high speed USB traces over continuous power planes, with no breaks. If a trade-off must be made, changing signal layers is preferable to crossing plane splits.
- Follow the $20 \times h$ rule; keep traces $20 \times h$ (the height above the power plane) away from the edge of the power plane.
- Use a minimum of vias in high speed USB traces.

A write message comprises the following:

control-token 192	24-bit response channel-end identifier	16-bit register number	32-bit 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 193	24-bit response channel-end identifier	16-bit register number	control-token 1

The response to the read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

A.3 Accessing digital and analogue node configuration registers

Node configuration registers can be accessed through the interconnect using the functions `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ...)`, where `device` is the name of the node. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the node configuration registers. The destination of the channel-end should be set to `0xnnnnC30C` where `nnnn` is the node-identifier.

A write message comprises the following:

control-token 192	24-bit response channel-end identifier	16-bit register number	32-bit 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 193	24-bit response channel-end identifier	16-bit register number	control-token 1

The response to a read message comprises either control token 3, 32-bit of data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

A.4 Accessing a register of an analogue peripheral

Peripheral registers can be accessed through the interconnect using the functions `write_periph_32(device, peripheral, ...)`, `read_periph_32(device, peripheral, ...)`, `write_periph_8(device, peripheral, ...)`, and `read_periph_8(device, peripheral, ...)`; where device is the name of the analogue device, and peripheral is the number of the peripheral. These functions implement the protocols described below.

0x03: xCORE Tile boot status

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		xCORE tile number on the switch.
15:9	RO	-	Reserved
8	RO		Set to 1 if boot from OTP is enabled.
7:0	RO		The boot mode pins MODE0, MODE1, ..., specifying the boot frequency, boot source, etc.

B.5 Security configuration: 0x05

Copy of the security register as read from OTP.

0x05: Security configuration

Bits	Perm	Init	Description
31:0	RO		Value.

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 is stopped. 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	Set to 1 to enable the xCORE tile ring oscillators
0	RW	0	Set to 1 to enable the peripheral ring oscillators

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.

0x07: Ring Oscillator Value

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

0x50 .. 0x53: Data watchpoint address 1

Bits	Perm	Init	Description
31:0	DRW		Value.

B.23 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.24 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 logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:3	RO	-	Reserved
2	DRW	0	Set to 1 to enable breakpoints to be triggered on loads. Breakpoints always trigger on stores.
1	DRW	0	By default, data watchpoints trigger if memory in the range [Address1..Address2] is accessed (the range is inclusive of Address1 and Address2). If set to 1, data watchpoints trigger if memory outside the range (Address2..Address1) is accessed (the range is exclusive of Address2 and Address1).
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.25 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.26 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.27 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 logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	By default, resource watchpoints trigger when the resource id masked with the set Mask equals the Value. If set to 1, resource watchpoints trigger when the resource id masked with the set Mask is not equal to the Value.
0	DRW	0	When 1 the instruction breakpoint is enabled.

C xCORE Tile Configuration

The xCORE Tile control registers can be accessed using configuration reads and writes (use `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tileref, ...)` for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	xCORE Tile description 1
0x02	RO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	RW	xCORE Tile clock divider
0x07	RO	Security configuration
0x10 .. 0x13	RO	PLink status
0x20 .. 0x27	CRW	Debug scratch
0x40	RO	PC of logical core 0
0x41	RO	PC of logical core 1
0x42	RO	PC of logical core 2
0x43	RO	PC of logical core 3
0x44	RO	PC of logical core 4
0x60	RO	SR of logical core 0
0x61	RO	SR of logical core 1
0x62	RO	SR of logical core 2
0x63	RO	SR of logical core 3
0x64	RO	SR of logical core 4
0x80 .. 0x9F	RO	Chanend status

Figure 45: Summary

C.1 Device identification: 0x00

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

0x00: Device identification

0x05: Cause debug interrupts

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RO	0	Set to 1 when the processor is in debug mode.
0	CRW	0	Set to 1 to 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

0x06: xCORE Tile clock divider

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	RW		Value of the clock divider minus one.

C.7 Security configuration: 0x07

Copy of the security register as read from OTP.

0x07: Security configuration

Bits	Perm	Init	Description
31:0	RO		Value.

C.8 PLink status: 0x10 .. 0x13

Status of each of the four processor links; connecting the xCORE tile to the switch.

0x10 .. 0x13: PLink status

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

C.9 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over the switch. This is the same set of registers as the Debug Scratch registers in the processor status.

0x20 .. 0x27: Debug scratch

Bits	Perm	Init	Description
31:0	CRW		Value.

C.10 PC of logical core 0: 0x40

Value of the PC of logical core 0.

0x40: PC of logical core 0

Bits	Perm	Init	Description
31:0	RO		Value.

C.11 PC of logical core 1: 0x41

0x41: PC of logical core 1

Bits	Perm	Init	Description
31:0	RO		Value.

C.12 PC of logical core 2: 0x42

0x42: PC of logical core 2

Bits	Perm	Init	Description
31:0	RO		Value.

C.13 PC of logical core 3: 0x43

0x43: PC of logical core 3

Bits	Perm	Init	Description
31:0	RO		Value.

C.14 PC of logical core 4: 0x44

0x44: PC of logical core 4

Bits	Perm	Init	Description
31:0	RO		Value.

C.15 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	RO		Value.

D.11 Debug source: 0x1F

Contains the source of the most recent debug event.

0x1F: Debug source

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

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

These registers contain status information for low level debugging (read-only), the network number that each link belongs to, and the direction that each link is part of. The registers control links C, D, A, B, G, H, E, and F in that order.

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

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

Bits	Perm	Init	Description
31	RW	0	Write '1' to this bit to enable the link, write '0' to disable it. This bit controls the muxing of ports with overlapping links.
30	RW	0	Set to 0 to operate in 2 wire mode or 1 to operate in 5 wire mode
29:28	RO	-	Reserved
27	RO	0	Set to 1 on error: an RX buffer overflow or illegal token encoding has been received. This bit clears on reading.
26	RO	0	1 if this end of the link has issued credit to allow the remote end to transmit.
25	RO	0	1 if this end of the link has credits to allow it to transmit.
24	WO	0	Set to 1 to initialize a half-duplex link. This clears this end of the link's credit and issues a HELLO token; the other side of the link will reply with credits. This bit is self-clearing.
23	WO	0	Set to 1 to reset the receiver. The next symbol that is detected will be assumed to be the first symbol in a token. This bit is self-clearing.
22	RO	-	Reserved
21:11	RW	0	The number of system clocks between two subsequent transitions within a token
10:0	RW	0	The number of system clocks between two subsequent transmit tokens.

0x80 .. 0x87: Link configuration and initialization

D.15 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.

Bits	Perm	Init	Description
31	RW	0	Enable static forwarding.
30:5	RO	-	Reserved
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.

0xA0 .. 0xA7: Static link configuration

Bits	Perm	Init	Description
31:25	RO	-	Reserved
24	RW		Tristate processor mode pins.
23:18	RO	-	Reserved
17:16	RW		Processor mode pins.
15:4	RO	-	Reserved
3	RW	0	USB peripheral register access enable.
2	RW	0	USB interface block enable. Set to 1 to enable. Set to 0 to disable and reset all USB interface registers
1	WO	0	xCORE Tile reset. Set to 1 to initiate a reset of the xCORE Tile. This bit is self clearing. A write to this configuration register with this bit asserted results in no response packet being sent to the sender regardless of whether or not a response was requested.
0	WO	0	System reset. Set to 1 to initiate a reset whose scope includes most configuration and peripheral control registers. This bit is self clearing. A write to this configuration register with this bit asserted results in no response packet being sent to the sender regardless of whether or not a response was requested.

0x50: Reset and Mode Control

E.5 System clock frequency: 0x51

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. The following functions depend on the correct frequency settings: * Processor reset delay * The watchdog clock * The real-time clock when running in sleep mode * The USB clock (USB requires a 12, 24, 48, or 96 MHz oscillator)

0x51: System clock frequency

E.6 Link Control and Status: 0x80

0x80: Link Control and Status

Bits	Perm	Init	Description
31:28	RO	-	Reserved
27	RO	0	Set to 1 on error: an RX buffer overflow or illegal token encoding has been received. This bit clears on reading.
26	RO	0	1 if this end of the link has issued credit to allow the remote end to transmit.
25	RO	0	1 if this end of the link has credits to allow it to transmit.
24	WO	0	Set to 1 to initialize a half-duplex link. This clears this end of the link's credit and issues a HELLO token; the other side of the link will reply with credits. This bit is self-clearing.
23	WO	0	Set to 1 to reset the receiver. The next symbol that is detected will be assumed to be the first symbol in a token. This bit is self-clearing.
22	RO	-	Reserved
21:11	RW	1	The number of system clocks between two subsequent transitions within a token
10:0	RW	1	The number of system clocks between two subsequent transmit tokens.

E.7 1 KHz Watchdog Control: 0xD6

The watchdog provides a mechanism to prevent programs from hanging by resetting the xCORE Tile after a pre-set time. The watchdog should be periodically "kicked" by the application, causing the count-down to be restarted. If the watchdog expires, it may be due to a program hanging, for example because of a (transient) hardware issue.

The watchdog timeout is measured in 1 ms clock ticks, meaning that a time between 1 ms and 65 seconds can be set for the timeout. The watchdog timer is only clocked during the AWAKE power state. When writing the timeout value, both the timeout and its one's complement should be written. This reduces the chances of accidentally setting kicking the watchdog. If the written value does not comprise a 16-bit value with a 16-bit one's complement, the request will be NACKed, otherwise an ACK will be sent.

If the watchdog expires, the xCORE Tile is reset.

0xD6: 1 KHz Watchdog Control

Bits	Perm	Init	Description
31:16	RO	0	Current value of watchdog timer.
15:0	RW	1000	Number of 1kHz cycles after which the watchdog should expire and initiate a system reset.

Number	Perm	Description
0x00	RW	General control
0x04	RW	Time to wake-up, least significant 32 bits
0x08	RW	Time to wake-up, most significant 32 bits
0x0C	RW	Power supply states whilst ASLEEP
0x10	RW	Power supply states whilst WAKING1
0x14	RW	Power supply states whilst WAKING2
0x18	RW	Power supply states whilst AWAKE
0x1C	RW	Power supply states whilst SLEEPING1
0x20	RW	Power supply states whilst SLEEPING2
0x24	RW	Power sequence status
0x2C	RW	DCDC control
0x30	RW	Power supply status
0x34	RW	VDDCORE level control
0x40	RW	LDO5 level control

Figure 53: Summary

K.1 General control: 0x00

This register controls the basic settings for power modes.

O PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS1-U10A-128-FB217. Each of the following sections contains items to check for each design.

O.1 Ground Balls and Ground Plane

☐ There is one via for every other ground ball to minimize impedance and conduct heat away from the device (Section 16.3).

☐ There are only few non-ground vias around the square of ground balls, to creating a good, solid, ground plane.

O.2 Power supply decoupling

☐ VSUP has a ceramic X5R or X7R bulk decoupler as close as possible to the VSUP and PGND (VDDCORE) pins; right next to the device (Section 16).

☐ The 1V0 decoupling cap is close to the VDDCORE and PGND pins (Section 16).

☐ The 1V8 decoupling cap is close to the VDD1V8 and PGND pins (Section 16).

☐ All PGND nets are connected together prior to connection to the main ground plane (Section 16).

An example PCB layout is shown in Section 17. Placing the decouplers too far away may lead to the device not coming up, or not operating properly.