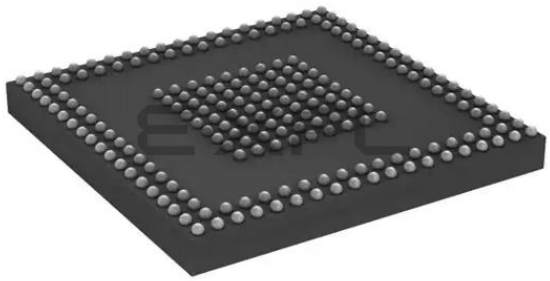

Welcome to E-XFL.COM

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 8-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-u8a-128-fb217-i10

1 xCORE Multicore Microcontrollers

The XS1-U 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.

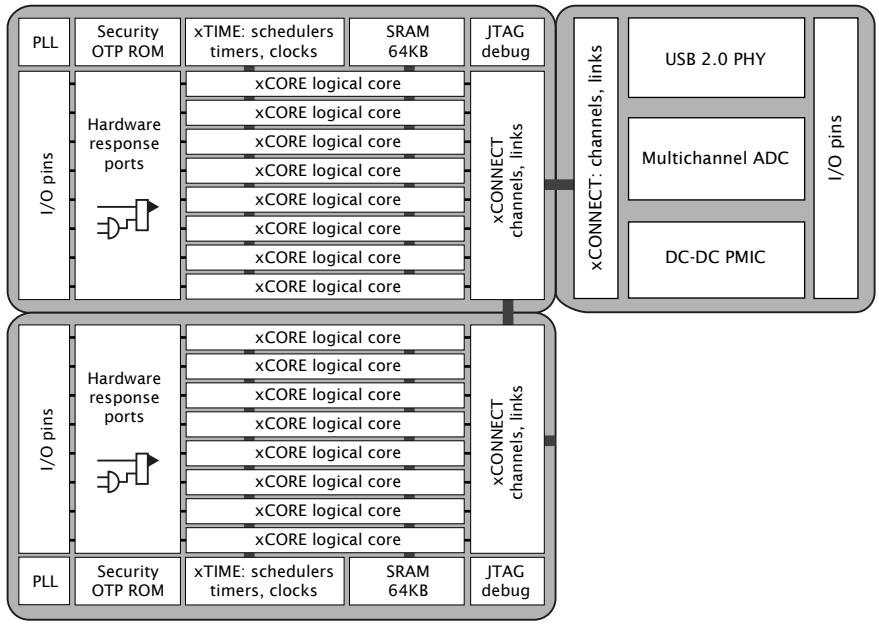

Figure 1: XS1-U Series: 6-16 core devices

Key features of the XS1-U8A-128-FB217 include:

- **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between four 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 7.1
- **xTIME scheduler** The xTIME scheduler performs functions similar to an RTOS, in hardware. It services and synchronizes events in a core, so there is no requirement for interrupt handler routines. The xTIME scheduler triggers cores on events generated by hardware resources such as the I/O pins, communication channels and timers. Once triggered, a core runs independently and concurrently to other cores, until it pauses to wait for more events. Section 7.2

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	4B X1D05	4B X1D06	4B X1D07	4A X1D08	4A X1D09	4A X1D10	4A X1D11	4A X1D12	4A X1D13	4A X1D14	4A X1D15	4A X1D16	4A X1D17	4A X1D18	4A X1D19	4A X1D20	4A X1D21	4A X1D22	4A X1D23
B	4B X1D04	4B X1D53	4B X1D54	4A X1D55	4A X1D56	4A X1D57	4A X1D58	4A X1D61	4A X1D62	4A X1D63	4A X1D64	4A X1D65	4A X1D66	4A X1D67	4A X1D68	4A X1D69	4A X1D70	4A X1D24	4A X1D25
C	4A X1D03	3A X1D52																4E X1D26	4E X1D27
D	4A X1D02	3A X1D51																4E X1D33	4E X1D32
E	4B X1D01	3A X1D50																1L X1D35	1K X1D34
F	1J X1D00	4A X1D49				GND	GND	GND	GND	GND	GND	GND	GND	GND				VDDIO_ OUT	1K X1D36
G	USB_ DN	USB VBUS				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[4]	1K X1D37
H	USB_ DP	USB_ ID				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[3]	1J X1D38
J	X0D43/ WAKE	RST_N				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[2]	1P 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	1N X0D00
T	NC	NC																NC	1P X0D01
U	XI/ CLK	NC																NC	1J 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

6 Product Overview

The XS1-U8A-128-FB217 comprises a digital and an analog node, as shown in Figure 3. The digital node comprises an xCORE Tile, a Switch, and a PLL (Phase-locked-loop). The analog node comprises the USB PHY, a multi-channel ADC (Analog to Digital Converter), deep sleep memory, an oscillator, a real-time counter, and power supply control.

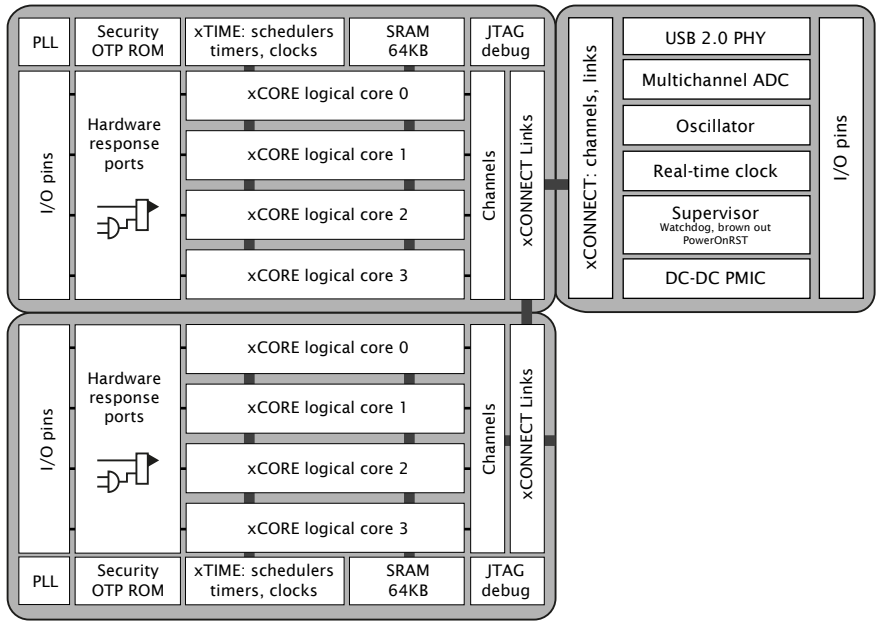

Figure 3: Block Diagram

All communication between the digital and analog node takes place over a link that is connected to the Switch of the digital node. As such, the analog node can be controlled from any node on the system. The analog functions can be configured using a set of node configuration registers, and a set of registers for each of the peripherals.

The device can be programmed using high-level languages such as C/C++ and the XMOS-originated XC language, which provides extensions to C that simplify the control over concurrency, I/O and timing, or low-level assembler.

6.1 XCore Tile

The xCORE Tile is a flexible multicore microcontroller component with tightly integrated I/O and on-chip memory. The tile contains multiple logical cores that run simultaneously, each of which is guaranteed a slice of processing power and can execute computational code, control software and I/O interfaces. The logical cores use channels to exchange data within a tile or across tiles. The tiles are connected via an integrated switch network, called xCONNECT, which uses a

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.
2. Input a word on channel-end 0. It will use this word as a channel to acknowledge the boot. Provide the null-channel-end 0x0000FF02 if no acknowledgment is required.
3. Input the boot image specified above, including the CRC.

18 DC and Switching Characteristics

18.1 Operating Conditions

Figure 26: Operating conditions

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VSUP	Power Supply (3.3V Mode)	3.00	3.30	3.60	V	
	Power Supply (5V Mode)	4.50	5.00	5.50	V	
VDDIO	I/O supply voltage	3.00	3.30	3.60	V	
AVDD	Analog Supply and Reference Voltage	3.00	3.30	3.60	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	

18.2 DC1 Characteristics

Figure 27: DC1 characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDDCORE	Tile Supply Voltage	0.95	1.00	1.05	V	
V(RIPPLE)	Ripple Voltage (peak to peak)		10	40	mV	
V(ACC)	Voltage Accuracy	-5		5	%	A
F(S)	Switching Frequency		1		MHz	
F(SVAR)	Variation in Switching Frequency	-10		10	%	
Effic	Efficiency		80		%	
PGT(HIGH)	Powergood Threshold (High)		95		%/VDDCORE	
PGT(LOW)	Powergood Threshold (Low)		80		%/VDDCORE	

A If supplied externally.

19.1 Part Marking

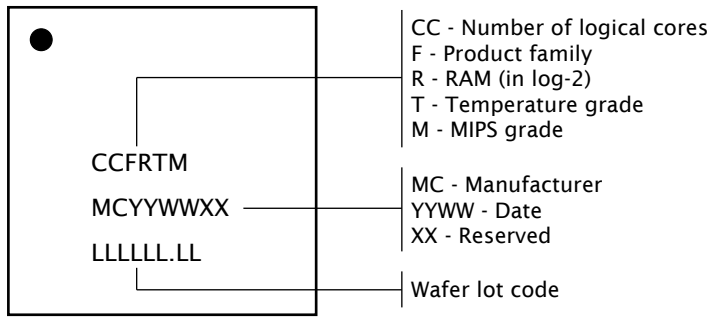

Figure 41: Part marking scheme

20 Ordering Information

Figure 42: Orderable part numbers

Product Code	Marking	Qualification	Speed Grade
XS1-U8A-128-FB217-C10	8U7C10	Commercial	1000 MIPS
XS1-U8A-128-FB217-I10	8U7I10	Industrial	1000 MIPS

A channel-end should be allocated to communicate with the configuration registers. The destination of the channel-end should be set to 0xnnnnpp02 where nnnn is the node-identifier and pp is the peripheral identifier.

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 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	RO	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
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 44: Summary

0x11: Debug SPC

Bits	Perm	Init	Description
31:0	DRW		Value.

B.13 Debug SSP: 0x12

This register contains the value of the SSP register when the debugger was called.

0x12: Debug SSP

Bits	Perm	Init	Description
31:0	DRW		Value.

B.14 DGETREG operand 1: 0x13

The resource ID of the logical core whose state is to be read.

0x13: DGETREG operand 1

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		Thread number to be read

B.15 DGETREG operand 2: 0x14

Register number to be read by DGETREG

0x14: DGETREG operand 2

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:0	DRW		Register number to be read

B.16 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

D Digital Node Configuration

The digital 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).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	System switch description
0x04	RW	Switch configuration
0x05	RW	Switch node identifier
0x06	RW	PLL settings
0x07	RW	System switch clock divider
0x08	RW	Reference clock
0x0C	RW	Directions 0-7
0x0D	RW	Directions 8-15
0x10	RW	DEBUG_N configuration
0x1F	RO	Debug source
0x20 .. 0x27	RW	Link status, direction, and network
0x40 .. 0x43	RW	PLink status and network
0x80 .. 0x87	RW	Link configuration and initialization
0xA0 .. 0xA7	RW	Static link configuration

Figure 46: Summary

D.1 Device identification: 0x00

This register contains version and revision identifiers and the mode-pins as sampled at boot-time.

Bits	Perm	Init	Description
31:24	RO	0x00	Chip identifier.
23:16	RO		Sampled values of pins MODE0, MODE1, ... on reset.
15:8	RO		SSwitch revision.
7:0	RO		SSwitch version.

0x00: Device identification

D.2 System switch description: 0x01

This register specifies the number of processors and links that are connected to this switch.

D.13 PLink status and network: 0x40 .. 0x43

These registers contain status information and the network number that each processor-link belongs to.

0x40 .. 0x43: PLink status 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.
15:6	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.

D.14 Link configuration and initialization: 0x80 .. 0x87

These registers contain configuration and debugging information specific to external links. The link speed and width can be set, the link can be initialized, and the link status can be monitored. The registers control links C, D, A, B, G, H, E, and F in that order.

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

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.

0x20: UIFM Sticky flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	Stickyness for each flag.

F.10 UIFM port masks: 0x24

Set of masks that identify how port 1N, port 1O and port 1P are affected by changes to the flags in FLAGS

0x24: UIFM port masks

Bits	Perm	Init	Description
31:23	RO	-	Reserved
22:16	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1P. If any flag listed in this bitmask is high, port 1P will be high.
15	RO	-	Reserved
14:8	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1O. If any flag listed in this bitmask is high, port 1O will be high.
7	RO	-	Reserved
6:0	RW	0	Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1N. If any flag listed in this bitmask is high, port 1N will be high.

F.11 UIFM SOF value: 0x28

USB Start-Of-Frame counter

0x28: UIFM SOF value

Bits	Perm	Init	Description
31:11	RO	-	Reserved
10:8	RW	0	Most significant 3 bits of SOF counter
7:0	RW	0	Least significant 8 bits of SOF counter

F.12 UIFM PID: 0x2C

The last USB packet identifier received

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 UIFM power signalling: 0x38

0x38: UIFM power signalling

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

G.1 ADC Control input pin 0: 0x00

Controls specific to ADC input pin 0.

0x00: ADC Control input pin 0

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.2 ADC Control input pin 1: 0x04

Controls specific to ADC input pin 1.

0x04: ADC Control input pin 1

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.3 ADC Control input pin 2: 0x08

Controls specific to ADC input pin 2.

0x08: ADC Control input pin 2

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.4 ADC Control input pin 3: 0x0C

Controls specific to ADC input pin 3.

0x0C: ADC Control input pin 3

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.5 ADC Control input pin 4: 0x10

Controls specific to ADC input pin 4.

0x10: ADC Control input pin 4

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.6 ADC Control input pin 5: 0x14

Controls specific to ADC input pin 5.

0x14: ADC Control input pin 5

Bits	Perm	Init	Description
31:8	RW	0	The node and channel-end identifier to which data for this ADC input pin should be send to. This is the top 24 bits of the channel-end identifier as allocated on an xCORE Tile.
7:1	RO	-	Reserved
0	RW	0	Set to 1 to enable this input pin on the ADC.

G.7 ADC Control input pin 6: 0x18

Controls specific to ADC input pin 6.

Figure 50: Summary

Number	Perm	Description
0x00 .. 0x7F	RW	Deep sleep memory
0xFF	RW	Deep sleep memory valid

H.1 Deep sleep memory: 0x00 .. 0x7F

128 bytes of memory that can be used to hold data when the xCORE Tile is powered down.

0x00 .. 0x7F: Deep sleep memory

Bits	Perm	Init	Description
7:0	RW		User defined data

H.2 Deep sleep memory valid: 0xFF

One byte of memory that is reset to 0. The program can write a non zero value in this register to indicate that the data in deep sleep memory is valid.

0xFF: Deep sleep memory valid

Bits	Perm	Init	Description
7:0	RW	0	User defined data, reset to 0.

I Oscillator Configuration

The *Oscillator* is peripheral 4. The control registers are accessed using 8-bit reads and writes (use `write_periph_8(device, 4, ...)` and `read_periph_8(device, 4, ...)` for reads and writes).

Figure 51: Summary

Number	Perm	Description
0x00	RW	General oscillator control
0x01	RW	On-silicon-oscillator control
0x02	RW	Crystal-oscillator control

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	0	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

0x20: Power supply states whilst SLEEPING2

K.10 Power sequence status: 0x24

This register defines the current status of the power supply controller.

P Associated Design Documentation

Document Title	Information	Document Number
Programming XC on XMOS Devices	Timers, ports, clocks, cores and channels	X9577
xTIMEcomposer User Guide	Compilers, assembler and linker/mapper Timing analyzer, xScope, debugger Flash and OTP programming utilities	X3766

Q Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	X7879
XS1 Port I/O Timing	Port timings	X5821
xCONNECT Architecture	Link, switch and system information	X4249
XS1-L Link Performance and Design Guidelines	Link timings	X2999
XS1-L Clock Frequency Control	Advanced clock control	X1433