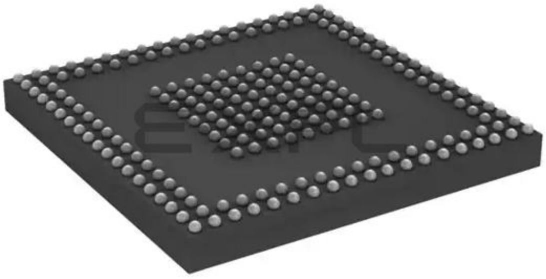

Welcome to E-XFL.COM

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	800MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	90
Program Memory Size	128KB (32K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	0.90V ~ 5.5V
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-a8a-128-fb217-i8

TO OUR VALUED CUSTOMERS

It is our intention to provide you with accurate and comprehensive documentation for the hardware and software components used in this product. To subscribe to receive updates, visit http://www.xmos.com/.

1 xCORE Multicore Microcontrollers

The XS1-A 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.

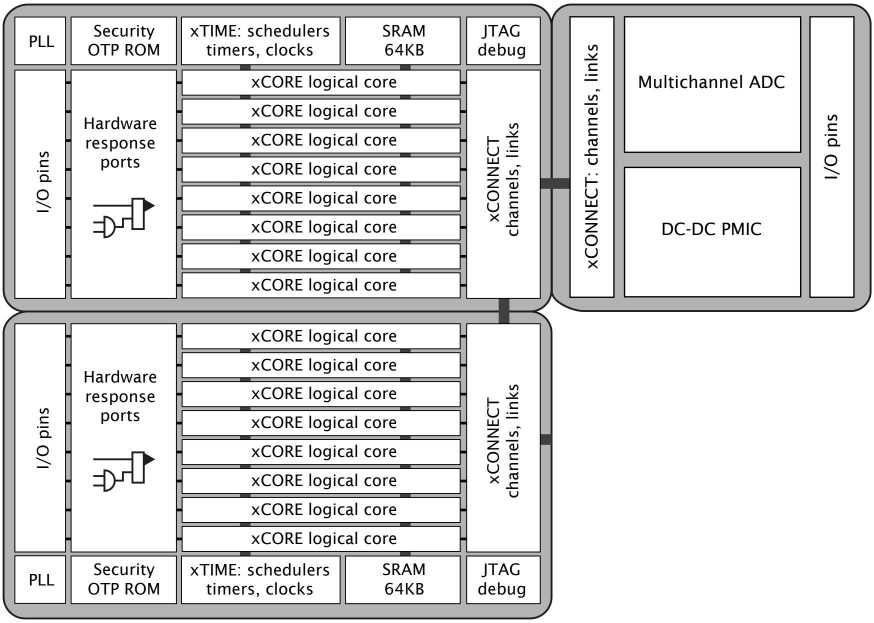

Figure 1: XS1-A Series:6-16 core devices

Key features of the XS1-A8A-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

2 XS1-A8A-128-FB217 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Eight real-time logical cores on 2 xCORE tiles
  - Cores share up to 1000 MIPS
  - Each logical core has:
    - Guaranteed throughput of 1/4 of tile MIPS
    - 16x32bit dedicated registers
  - 159 high-density 16/32-bit instructions
    - All have single clock-cycle execution (except for divide)
    - 32x32→64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions
- **12b 1MSPS 8-channel SAR Analog-to-Digital Converter**
- **1 x LDO**
- **2 x DC-DC converters and Power Management Unit**
- **Watchdog Timer**
- **Onchip clocks/oscillators**
  - Crystal oscillator
  - 20MHz/31kHz silicon oscillators
- **Programmable I/O**
  - 90 general-purpose I/O pins, configurable as input or output
    - Up to 32 x 1bit port, 12 x 4bit port, 8 x 8bit port, 4 x 16bit port
    - 4 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 64 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 128KB internal single-cycle SRAM (max 64KB per tile) for code and data storage
  - 16KB internal OTP (max 8KB per tile) for application boot code
  - 128 bytes Deep Sleep Memory
- **Hardware resources**
  - 12 clock blocks (6 per tile)
  - 20 timers (10 per tile)
  - 8 locks (4 per tile)
- **JTAG Module for On-Chip Debug**
- **Security Features**
  - Programming lock disables debug and prevents read-back of memory contents
  - AES bootloader ensures secrecy of IP held on external flash memory
- **Ambient Temperature Range**
  - Commercial qualification: 0 °C to 70 °C
  - Industrial qualification: -40 °C to 85 °C
- **Speed Grade**
  - 10: 1000 MIPS
  - 8: 800 MIPS
- **Power Consumption (typical)**
  - 600 mW at 500 MHz (typical)
  - Sleep Mode: 500 µW
- **217-pin FBGA package 0.8 mm pitch**

Signal	Function						Type	Properties
X1D15	XLB^{2}_{out}		$4C^{1}$	$8B^{1}$	$16A^{9}$	$32A^{29}$	I/O	PD_S, R_U
X1D16	XLB^{1}_{out}		$4D^{0}$	$8B^{2}$	$16A^{10}$		I/O	PD_S, R_U
X1D17	XLB^{0}_{out}		$4D^{1}$	$8B^{3}$	$16A^{11}$		I/O	PD_S, R_U
X1D18	XLB^{0}_{in}		$4D^{2}$	$8B^{4}$	$16A^{12}$		I/O	PD_S, R_U
X1D19	XLB^{1}_{in}		$4D^{3}$	$8B^{5}$	$16A^{13}$		I/O	PD_S, R_U
X1D20	XLB^{2}_{in}		$4C^{2}$	$8B^{6}$	$16A^{14}$	$32A^{30}$	I/O	PD_S, R_U
X1D21	XLB^{3}_{in}		$4C^{3}$	$8B^{7}$	$16A^{15}$	$32A^{31}$	I/O	PD_S, R_U
X1D22	XLB^{4}_{in}	$1G^{0}$					I/O	PD_S, R_U
X1D23		$1H^{0}$					I/O	PD_S, R_U
X1D24		$1I^{0}$					I/O	PD_S
X1D25		$1J^{0}$					I/O	PD_S
X1D26			$4E^{0}$	$8C^{0}$	$16B^{0}$		I/O	PD_S, R_U
X1D27			$4E^{1}$	$8C^{1}$	$16B^{1}$		I/O	PD_S, R_U
X1D28			$4F^{0}$	$8C^{2}$	$16B^{2}$		I/O	PD_S, R_U
X1D29			$4F^{1}$	$8C^{3}$	$16B^{3}$		I/O	PD_S, R_U
X1D30			$4F^{2}$	$8C^{4}$	$16B^{4}$		I/O	PD_S, R_U
X1D31			$4F^{3}$	$8C^{5}$	$16B^{5}$		I/O	PD_S, R_U
X1D32			$4E^{2}$	$8C^{6}$	$16B^{6}$		I/O	PD_S, R_U
X1D33			$4E^{3}$	$8C^{7}$	$16B^{7}$		I/O	PD_S, R_U
X1D34		$1K^{0}$					I/O	PD_S
X1D35		$1L^{0}$					I/O	PD_S
X1D36		$1M^{0}$		$8D^{0}$	$16B^{8}$		I/O	PD_S
X1D37		$1N^{0}$		$8D^{1}$	$16B^{9}$		I/O	PD_S, R_U
X1D38		$1O^{0}$		$8D^{2}$	$16B^{10}$		I/O	PD_S, R_U
X1D39		$1P^{0}$		$8D^{3}$	$16B^{11}$		I/O	PD_S, R_U
X1D40				$8D^{4}$	$16B^{12}$		I/O	PD_S, R_U
X1D41				$8D^{5}$	$16B^{13}$		I/O	PD_S, R_U
X1D42				$8D^{6}$	$16B^{14}$		I/O	PD_S, R_U
X1D43				$8D^{7}$	$16B^{15}$		I/O	PU_S, R_U
X1D70						$32A^{19}$	I/O	PD_S

frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.

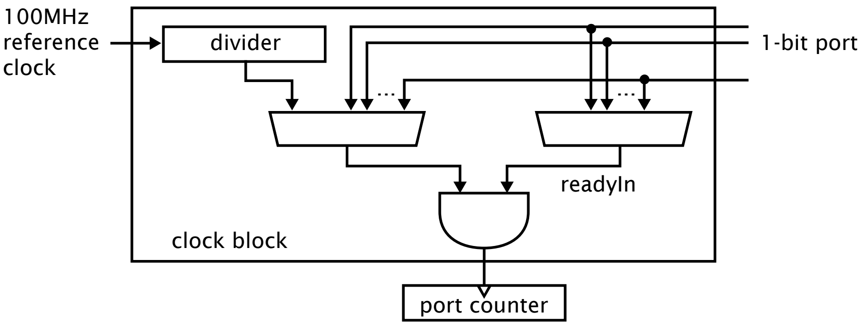

Figure 6: Clock block diagram

A clock block can use a 1-bit port as its clock source allowing external application clocks to be used to drive the input and output interfaces.

In many cases I/O signals are accompanied by strobing signals. The xCORE ports can input and interpret strobe (known as readyIn and readyOut) signals generated by external sources, and ports can generate strobe signals to accompany output data.

On reset, each port is connected to clock block 0, which runs from the processor reference clock.

7.5 Channels and Channel Ends

Logical cores communicate using point-to-point connections, formed between two channel ends. A channel-end is a resource on an xCORE tile, that is allocated by the program. Each channel-end has a unique system-wide identifier that comprises a unique number and their tile identifier. Data is transmitted to a channel-end by an output-instruction; and the other side executes an input-instruction. Data can be passed synchronously or asynchronously between the channel ends.

7.6 xCONNECT Switch and Links

XMOS devices provide a scalable architecture, where multiple xCORE devices can be connected together to form one system. Each xCORE device has an xCONNECT interconnect that provides a communication infrastructure for all tasks that run on the various xCORE tiles on the system.

The interconnect relies on a collection of switches and XMOS links. Each xCORE device has an on-chip switch that can set up circuits or route data. The switches are connected by xConnect Links. An XMOS link provides a physical connection between two switches. The switch has a routing algorithm that supports many different topologies, including lines, meshes, trees, and hypercubes.

The links operate in either 2 wires per direction or 5 wires per direction mode, depending on the amount of bandwidth required. Circuit switched, streaming

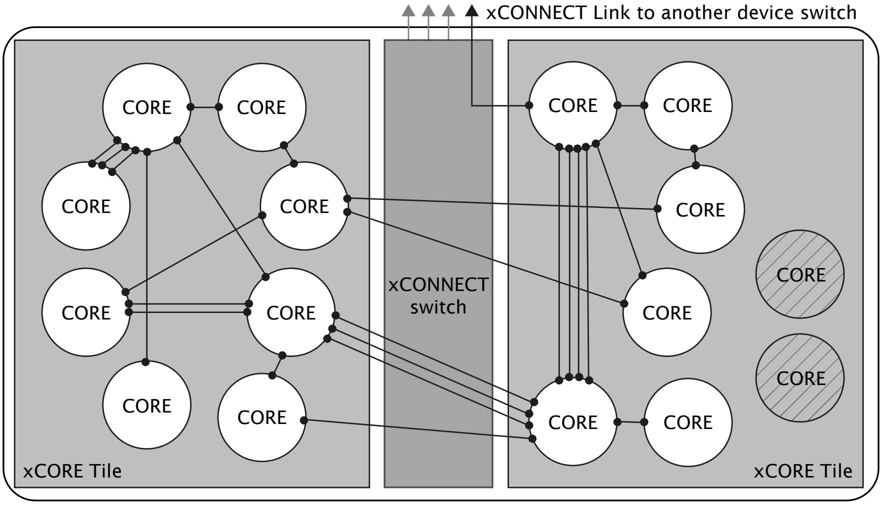

Figure 7: Switch, links and channel ends

and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between tiles , but each stream requires a single link to be reserved between switches on two tiles. All packet communications can be multiplexed onto a single link.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-L Link Performance and Design Guide, X2999.

8 Oscillator

The oscillator block provides:

- An oscillator circuit. Together with an external resonator (crystal or ceramic), the oscillator circuit can provide a clock-source for both the real-time counter and the xCORE Tile. The external resonator can be chosen by the designer to have the appropriate frequency and accuracy. If desired, an external oscillator can be used on the XI/CLK input pin, this must be a 1.8 V oscillator.
- A 20 MHz silicon oscillator. This enables the device to boot and execute code without requiring an external crystal. The silicon oscillator is not as accurate as an external crystal.
- A 31,250 Hz oscillator. This enables the real-time counter to operate whilst the device is in low-power mode. This oscillator is not as accurate as an external crystal.

The oscillator can be controlled through package pins, a set of peripheral registers, and a digital node control register.

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.
4. Input an END control token.
5. Output an END control token to the channel-end received in step 2.
6. Free channel-end 0.
7. Jump to the loaded code.

17.5 Digital I/O Characteristics

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	2.00			V	B, C
V(OL)	Output low voltage			0.60	V	B, C
R(PU)	Pull-up resistance		35K		Ω	D
R(PD)	Pull-down resistance		35K		Ω	D

Figure 28: Digital I/O characteristics

A All pins except power supply pins.

B Ports 1A, 1D, 1E, 1H, 1I, 1J, 1K and 1L 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.

17.6 ESD Stress Voltage

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
HBM	Human body model			2.00	kV	
CDM	Charged Device Model			500	V	

Figure 29: ESD stress voltage

17.7 Device Timing Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
T(RST)	Reset pulse width	5			µs	
T(INIT)	Initialisation (On Silicon Oscillator)			TBC	ms	A
	Initialisation (Crystal Oscillator)			TBC	ms	
T(WAKE)	Wake up time (Sleep to Active)			TBC	ms	
T(SLEEP)	Sleep Time (Active to Sleep)			TBC	ms	

Figure 30: Device timing characteristics

A Shows the time taken to start booting after RST_N has gone high.

17.8 Crystal Oscillator Characteristics

Figure 31: Crystal oscillator characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
F(FO)	Input Frequency	5		30	MHz	

17.9 External Oscillator Characteristics

Figure 32: External oscillator characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
F(EXT)	External Frequency			100	MHz	
V(IH)	Input high voltage	1.62		1.98	V	
V(IL)	Input low voltage			0.4	V	

17.10 Power Consumption

Figure 33: xCORE Tile currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
P(AWAKE)	Active Power for awake states (Speed Grade 10)	TBC	600	TBC	mW	
	Active Power for awake states (Speed Grade 8)	TBC	480	TBC	mW	
P(SLEEP)	Power when asleep	TBC	500	TBC	µW	

17.11 Clock

Figure 34: Clock

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(MAX)	Processor clock frequency (Speed Grade 10)			500	MHz	A
	Processor clock frequency (Speed Grade 8)			400	MHz	A

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

17.12 Processor I/O AC Characteristics

Figure 35: 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	

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.

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.

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.

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 43: 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.

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.

F.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.

F.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.

F.4 ADC Control input pin 3: 0x0C

Controls specific to ADC input pin 3.

0x18: ADC Control input pin 6

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.

F.8 ADC Control input pin 7: 0x1C

Controls specific to ADC input pin 7.

0x1C: ADC Control input pin 7

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.

F.9 ADC General Control: 0x20

General ADC control.

0x08: Time to wake-up, most significant 32 bits

Bits	Perm	Init	Description
31:0	RW	0	Most significant 32 bits of time to wake-up (ignored unless 64-bit timer comparison is enabled).

J.4 Power supply states whilst ASLEEP: 0x0C

This register controls the state the power control block should be in when in the ASLEEP state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state may be entered if either of the wake conditions (real-time counter or WAKE pin) happens. Note that the minimum number of cycles is counted in according to the currently enabled clock, which may be the slow 31 KHz clock.

0x0C: Power supply states whilst ASLEEP

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	0	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

0x18: Power supply states whilst AWAKE

Bits	Perm	Init	Description
31:15	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	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	1	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	1	Set to 1 to enable DCDC1 (core supply).

J.8 Power supply states whilst SLEEPING1: 0x1C

This register controls what state the power control block should be in when in the SLEEPING1 state. It also defines the time that the system shall stay in this state.

0x20: Power supply states whilst SLEEPING2

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).

J.10 Power sequence status: 0x24

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

0x24: Power sequence status

Bits	Perm	Init	Description
31:30	RO	-	Reserved
29	RO	0	1 if VOUT6 was enabled in the previous state.
28	RO	0	1 if LDO5 was enabled in the previous state.
27:26	RO	-	Reserved
25	RO	1	1 if DCDC2 was enabled in the previous state.
24	RO	0	1 if DCDC1 was enabled in the previous state.
23:19	RO	-	Reserved
18:16	RO		Current state of the power sequence state machine 0: Reset 1: Asleep 2: Waking 1 3: Waking 2 4: Awake Wait 5: Awake 6: Sleeping 1 7: Sleeping 2
15	RO	-	Reserved
14	RO	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RO	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RO	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RO	0	Set to 1 to enable VOUT6 (IO supply).
4	RO	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	0	Set to 1 to enable DCDC2 (analogue supply).
0	RO	0	Set to 1 to enable DCDC1 (core supply).

J.11 DCDC control: 0x2C

This register controls the two DC-DC converters.

K XMOS USB Interface

XMOS provides a low-level USB interface for connecting the device to a USB transceiver using the UTMI+ Low Pin Interface (ULPI). The ULPI signals must be connected to the pins named in Figure 50. Note also that some ports on the same tile are used internally and are not available for use when the USB driver is active (they are available otherwise).

Pin	Signal
X*n*D02	Unavailable when USB active
X*n*D03	
X*n*D04	
X*n*D05	
X*n*D06	
X*n*D07	
X*n*D08	
X*n*D09	

Pin	Signal
X*n*D12	ULPI_STP
X*n*D13	ULPI_NXT
X*n*D14	ULPI_DATA[0]
X*n*D15	ULPI_DATA[1]
X*n*D16	ULPI_DATA[2]
X*n*D17	ULPI_DATA[3]
X*n*D18	ULPI_DATA[4]
X*n*D19	ULPI_DATA[5]
X*n*D20	ULPI_DATA[6]
X*n*D21	ULPI_DATA[7]
X*n*D22	ULPI_DIR
X*n*D23	ULPI_CLK

Pin	Signal
X*n*D26	Unavailable when USB active
X*n*D27	
X*n*D28	
X*n*D29	
X*n*D30	
X*n*D31	
X*n*D32	
X*n*D33	

X*n*D37	Unavailable when USB active
X*n*D38	
X*n*D39	
X*n*D40	
X*n*D41	
X*n*D42	
X*n*D43	

Figure 50: ULPI signals provided by the XMOS USB driver

L Device Errata

This section describes minor operational differences from the data sheet and recommended workarounds. As device and documentation issues become known, this section will be updated the document revised.

To guarantee a logic low is seen on the pins DEBUG_N, MODE[4:0], TMS, TCK and TDI, the driving circuit should present an impedance of less than 100 Ω to ground. Usually this is not a problem for CMOS drivers driving single inputs. If one or more of these inputs are placed in parallel, however, additional logic buffers may be required to guarantee correct operation.

For static inputs tied high or low, the relevant input pin should be tied directly to GND or VDDIO.

M JTAG, xSCOPE and Debugging

If you intend to design a board that can be used with the XMOS toolchain and xTAG debugger, you will need an xSYS header on your board. Figure 51 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

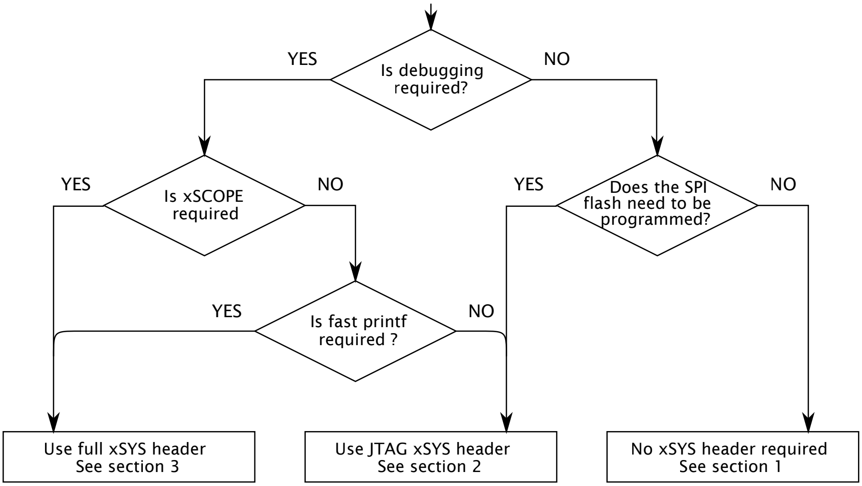

Figure 51: Decision diagram for the xSYS header

M.1 No xSYS header

The use of an xSYS header is optional, and may not be required for volume production designs. However, the XMOS toolchain expects the xSYS header; if you do not have an xSYS header then you must provide your own method for writing to flash/OTP and for debugging.

M.2 JTAG-only xSYS header

The xSYS header connects to an xTAG debugger, which has a 20-pin 0.1" female IDC header. The design will hence need a male IDC header. We advise to use a boxed header to guard against incorrect plug-ins. If you use a 90 degree angled header, make sure that pins 2, 4, 6, ..., 20 are along the edge of the PCB.

Connect pins 4, 8, 12, 16, 20 of the xSYS header to ground, and then connect:

- TDI to pin 5 of the xSYS header
- TMS to pin 7 of the xSYS header
- TCK to pin 9 of the xSYS header
- DEBUG_N to pin 11 of the xSYS header