Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of "Quality Parts,Customers Priority,Honest Operation,and Considerate Service",our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

XS1-U8A-128-FB217 Datasheet

Table of Contents

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/.

XMOS Ltd. is the owner or licensee of the information in this document and is providing it to you "AS IS" with no warranty of any kind, express or implied and shall have no liability in relation to its use. XMOS Ltd. makes no representation that the information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

XMOS and the XMOS logo are registered trademarks of XMOS Ltd in the United Kingdom and other countries, and may not be used without written permission. Company and product names mentioned in this document are the trademarks or registered trademarks of their respective owners.

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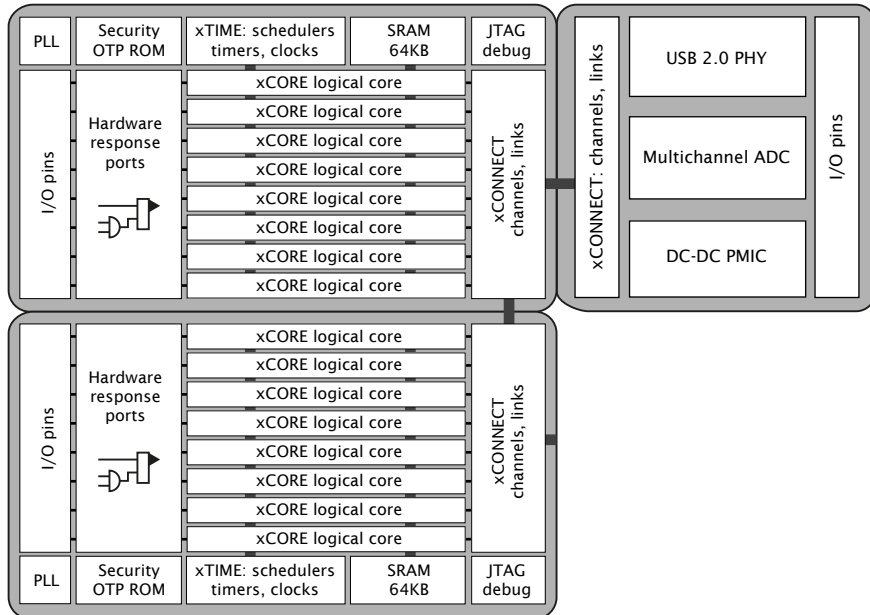
Figure 1:
XS1-U Series:
6-16 core
devices

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

▶ **Channels and channel ends** Tasks running on logical cores communicate using channels formed between two channel ends. Data can be passed synchronously or asynchronously between the channel ends assigned to the communicating tasks. Section 7.5

▶ **xCONNECT Switch and Links** Between tiles, channel communications are implemented over a high performance network of xCONNECT Links and routed through a hardware xCONNECT Switch. Section 7.6

▶ **Ports** The I/O pins are connected to the processing cores by Hardware Response ports. The port logic can drive its pins high and low, or it can sample the value on its pins optionally waiting for a particular condition. Section 7.3

▶ **Clock blocks** xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 7.4

▶ **Memory** Each xCORE Tile integrates a bank of SRAM for instructions and data, and a block of one-time programmable (OTP) memory that can be configured for system wide security features. Section 10

▶ **PLL** The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 8

▶ **USB** The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 11

▶ **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 15

1.1 Software

Devices are programmed using C, C++ or xC (C with multicore extensions). XMOS provides tested and proven software libraries, which allow you to quickly add interface and processor functionality such as USB, Ethernet, PWM, graphics driver, and audio EQ to your applications.

1.2 xTIMEcomposer Studio

The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: uniquely in the embedded world, xTIMEcomposer Studio therefore includes a static timing analyzer, cycle-accurate simulator, and high-speed in-circuit instrumentation.

xTIMEcomposer can be driven from either a graphical development environment, or the command line. The tools are supported on Windows, Linux and MacOS X and available at no cost from xmos.com/downloads. Information on using the tools is provided in the xTIMEcomposer User Guide, X3766.

2 XS1-U8A-128-FB217 Features

▶ **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Eight real-time logical cores on 2 xCORE tiles
  - Cores share up to 500 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

▶ **USB PHY, fully compliant with USB 2.0 specification**

▶ **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**
  - 73 general-purpose I/O pins, configurable as input or output
    — Up to 25 x 1bit port, 7 x 4bit port, 3 x 8bit port, 1 x 16bit port, 1 x 32bit port
    — 5 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
  - 8KB 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

▶ **Power Consumption with USB running (typical)**
  - 600 mW (typical)
  - Sleep Mode: 500 µW

▶ **217-pin BG217 package 0.8 mm pitch**

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	X1D05	X1D06	X1D07	X1D08	X1D09	X1D10	X1D11	X1D12	X1D13	X1D14	X1D15	X1D16	X1D17	X1D18	X1D19	X1D20	X1D21	X1D22	X1D23
B	X1D04	X1D53	X1D54	X1D55	X1D56	X1D57	X1D58	X1D61	X1D62	X1D63	X1D64	X1D65	X1D66	X1D67	X1D68	X1D69	X1D70	X1D24	X1D25
C	X1D03	X1D52																X1D26	X1D27
D	X1D02	X1D51																X1D33	X1D32
E	X1D01	X1D50																X1D35	X1D34
F	X1D00	X1D49				GND	GND	GND	GND	GND	GND	GND	GND	GND				VDDIO_OUT	X1D36
G	USB_DN	USB_VBUS				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[4]	X1D37
H	USB_DP	USB_ID				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[3]	X1D38
J	X0D43/WAKE	RST_N				GND	GND	GND	GND	GND	GND	GND	GND	GND				MODE[2]	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	X0D35
R	ADC0	ADC1																NC	X0D00
T	NC	NC																NC	X0D01
U	XI/CLK	NC																NC	X0D10
V	XO	NC	VDDCORE	PGND	PGND	SW1	VSUP	VDD1V8	PGND	PGND	SW2	NC	X0D24	X0D21	X0D19	X0D17	X0D15	NC	X0D11
W	VSUP	NC	VDDCORE	VDDCORE	PGND	SW1	VSUP	VDD1V8	VDD1V8	PGND	SW2	NC	X0D22	X0D20	X0D18	X0D16	X0D14	X0D13	X0D12

4 Signal Description

This section lists the signals and I/O pins available on the XS1-U8A-128-FB217. The device provides a combination of 1bit, 4bit, 8bit and 16bit ports, as well as wider ports that are fully or partially (gray) bonded out. All pins of a port provide either output or input, but signals in different directions cannot be mapped onto the same port.

Pins may have one or more of the following properties:

▶ PD/PU: The IO pin a weak pull-down or pull-up resistor. On GPIO pins this resistor can be enabled.

▶ ST: The IO pin has a Schmitt Trigger on its input.

Power pins (10)			
Signal	**Function**	**Type**	**Properties**
AVSS	Digital ground	GND	
GND	Digital ground	GND	
PGND	Power ground	GND	
SW1	DCDC1 switched output voltage	PWR	
SW2	DCDC2 switched output voltage	PWR	
VDD1V8	1v8 voltage supply	PWR	
VDDCORE	Core voltage supply	PWR	
VDDIO	Digital I/O power	PWR	
VDDIO_OUT	Digital I/O power out	PWR	
VSUP	Power supply (3V3/5V0)	PWR	

Analog pins (9)			
Signal	**Function**	**Type**	**Properties**
ADC0	Analog input	Input	
ADC1	Analog input	Input	
ADC2	Analog input	Input	
ADC3	Analog input	Input	
ADC4	Analog input	Input	
ADC5	Analog input	Input	
ADC6	Analog input	Input	
ADC7	Analog input	Input	
AVDD	Supply and reference voltage	PWR	

USB pins (4)			
Signal	**Function**	**Type**	**Properties**
USB_DN	USB Serial Data Inverted	I/O	

(continued)

Signal	Function	Type	Properties
USB_DP	USB Serial Data	I/O	
USB_ID	USB Device ID (OTG) - Reserved	Output	
USB_VBUS	USB Power Detect Pin	Input	

Clocks pins (4)			
Signal	**Function**	**Type**	**Properties**
MODE[4:0]	Boot mode select	Input	PU, ST
OSC_EXT_N	Use Silicon Oscillator	Input	ST
XI/CLK	Crystal Oscillator/Clock Input	Input	
XO	Crystal Oscillator Output	Output	

JTAG pins (5)			
Signal	**Function**	**Type**	**Properties**
DEBUG_N	Multi-chip debug	I/O	PU
TCK	Test clock	Input	PU, ST
TDI	Test data input	Input	PU, ST
TDO	Test data output	Output	PD, OT
TMS	Test mode select	Input	PU, ST

Misc pins (1)			
Signal	**Function**	**Type**	**Properties**
RST_N	Global reset input	Input	PU, ST

I/O pins (74)						
Signal	**Function**				**Type**	**Properties**
X0D00	$1A^0$				I/O	PD_S, R_S
X0D01	$1B^0$				I/O	PD_S, R_S
X0D10	$1C^0$				I/O	PD_S, R_S
X0D11	$1D^0$				I/O	PD_S, R_S
X0D12	$1E^0$				I/O	PD_S
X0D13	XLB_{out}^4	$1F^0$			I/O	PD_S
X0D14	XLB_{out}^3	$4C^0$	$8B^0$	$16A^8$ $32A^{28}$	I/O	PD_S
X0D15	XLB_{out}^2	$4C^1$	$8B^1$	$16A^9$ $32A^{29}$	I/O	PD_S
X0D16	XLB_{out}^1	$4D^0$	$8B^2$	$16A^{10}$	I/O	PD_S
X0D17	XLB_{out}^0	$4D^1$	$8B^3$	$16A^{11}$	I/O	PD_S
X0D18	XLB_{in}^0	$4D^2$	$8B^4$	$16A^{12}$	I/O	PD_S
X0D19	XLB_{in}^1	$4D^3$	$8B^5$	$16A^{13}$	I/O	PD_S
X0D20	XLB_{in}^2	$4C^2$	$8B^6$	$16A^{14}$ $32A^{30}$	I/O	PD_S

(continued)

Signal	Function						Type	Properties
X0D21	XLB_{in}^3		$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	PD_S
X0D22	XLB_{in}^4	$1G^0$					I/O	PD_S
X0D24		$1I^0$					I/O	PD_S
X0D35		$1L^0$					I/O	PD_S
X0D43/WAKE				$8D^7$	$16B^{15}$		I/O	PU_S
X1D00		$1A^0$					I/O	PD_S, R_S
X1D01	XLA_{out}^4	$1B^0$					I/O	PD_S, R_S
X1D02	XLA_{out}^3		$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	PD_S
X1D03	XLA_{out}^2		$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	I/O	PD_S
X1D04	XLA_{out}^1		$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O	PD_S
X1D05	XLA_{out}^0		$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O	PD_S
X1D06	XLA_{in}^0		$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O	PD_S
X1D07	XLA_{in}^1		$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O	PD_S
X1D08	XLA_{in}^2		$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	PD_S
X1D09	XLA_{in}^3		$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	PD_S
X1D10	XLA_{in}^4	$1C^0$					I/O	PD_S, R_S
X1D11		$1D^0$					I/O	PD_S, R_S
X1D12		$1E^0$					I/O	PD_S
X1D13	XLB_{out}^4	$1F^0$					I/O	PD_S
X1D14	XLB_{out}^3		$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	PD_S
X1D15	XLB_{out}^2		$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	PD_S
X1D16	XLB_{out}^1		$4D^0$	$8B^2$	$16A^{10}$		I/O	PD_S
X1D17	XLB_{out}^0		$4D^1$	$8B^3$	$16A^{11}$		I/O	PD_S
X1D18	XLB_{in}^0		$4D^2$	$8B^4$	$16A^{12}$		I/O	PD_S
X1D19	XLB_{in}^1		$4D^3$	$8B^5$	$16A^{13}$		I/O	PD_S
X1D20	XLB_{in}^2		$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	PD_S
X1D21	XLB_{in}^3		$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	PD_S
X1D22	XLB_{in}^4	$1G^0$					I/O	PD_S
X1D23		$1H^0$					I/O	PD_S
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
X1D27			$4E^1$	$8C^1$	$16B^1$		I/O	PD_S
X1D32			$4E^2$	$8C^6$	$16B^6$		I/O	PD_S
X1D33			$4E^3$	$8C^7$	$16B^7$		I/O	PD_S
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
X1D38		$1O^0$		$8D^2$	$16B^{10}$		I/O	PD_S
X1D39		$1P^0$		$8D^3$	$16B^{11}$		I/O	PD_S
X1D49	XLC_{out}^4					$32A^0$	I/O	PD_S
X1D50	XLC_{out}^3					$32A^1$	I/O	PD_S

(continued)

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

5 Example Application Diagram
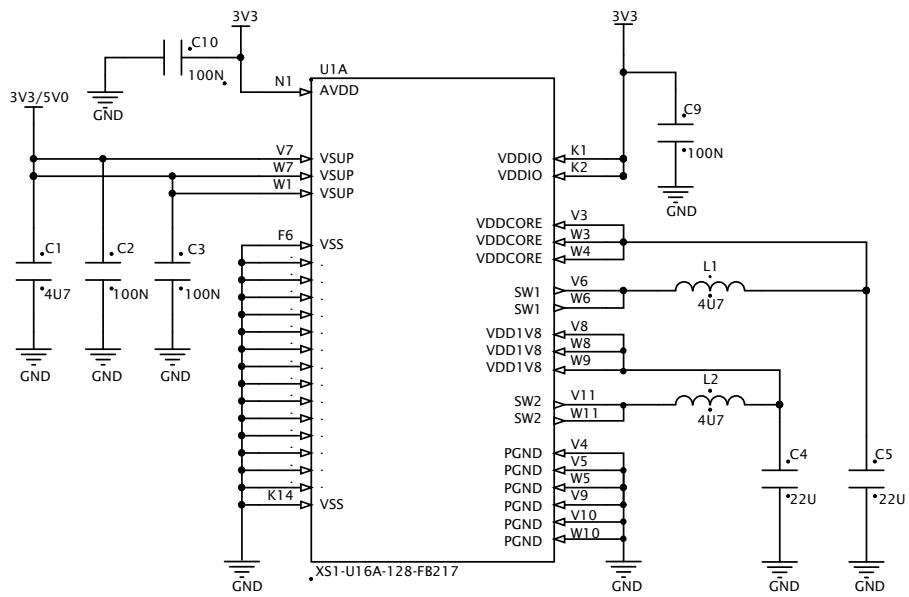
6   Product Overview

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.
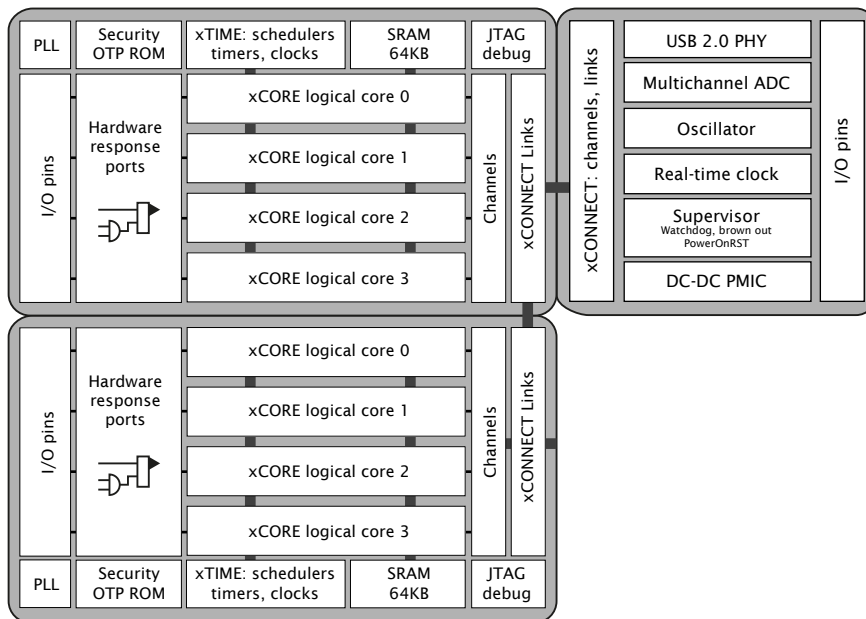

Figure 3:
Block
Diagram

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

proprietary physical layer protocol and can also be used to add additional resources to a design. The I/O pins are driven using intelligent ports that can serialize data, interpret strobe signals and wait for scheduled times or events, making the device ideal for real-time control applications.

6.2 USB PHY

The USB PHY is fully compliant with the USB 2.0 specification. It supports high speed (480-Mbps) and full speed (12Mbps) operation.

The XMOS XUD software component performs all the low-level I/O operations required to meet the USB 2.0 specification, removing all low-level timing requirements from the application.

6.3 ADC and Power Management

Each XS1-U8A-128-FB217 device includes a set of analog components, including a 12b, 8-channel ADC, power management unit, watchdog timer, real-time counter and deep sleep memory. The device reduces the number of additional external components required and allows designs to be implemented using simple 2-layer boards.

7 xCORE Tile Resources

7.1 Logical cores

Each tile has up to 4 active logical cores, which issue instructions down a shared four-stage pipeline. Instructions from the active cores are issued round-robin. Each core is allocated a quarter of the processing cycles. Figure 4 shows the guaranteed core performance.

Figure 4:
Logical core
performance

Speed grade	MIPS	Frequency	MIPS per logical core
10	1000 MIPS	500 MHz	125

There is no way that the performance of a logical core can be reduced below these predicted levels.

The logical cores are triggered by events instead of interrupts and run to completion. A logical core can be paused to wait for an event.

7.2 xTIME scheduler

The xTIME scheduler handles the events generated by xCORE Tile resources, such as channel ends, timers and I/O pins. It ensures that all events are serviced and synchronized, without the need for an RTOS. Events that occur at the I/O pins are handled by the Hardware-Response ports and fed directly to the appropriate xCORE

Tile. An xCORE Tile can also choose to wait for a specified time to elapse, or for data to become available on a channel.

Tasks do not need to be prioritised as each of them runs on their own logical xCORE. It is possible to share a set of low priority tasks on a single core using cooperative multitasking.

7.3 Hardware Response Ports

Hardware Response ports connect an xCORE tile to one or more physical pins and as such define the interface between hardware attached to the XS1-U8A-128-FB217, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.
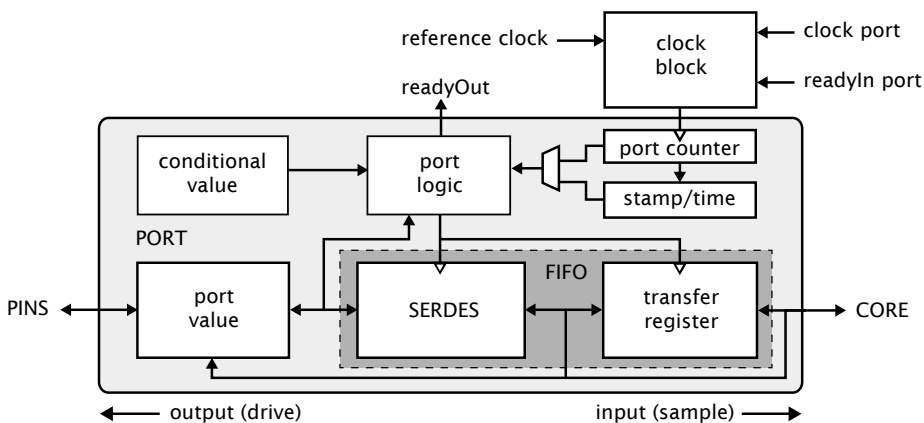
Figure 5:
Port block diagram

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.

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.

7.4   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.
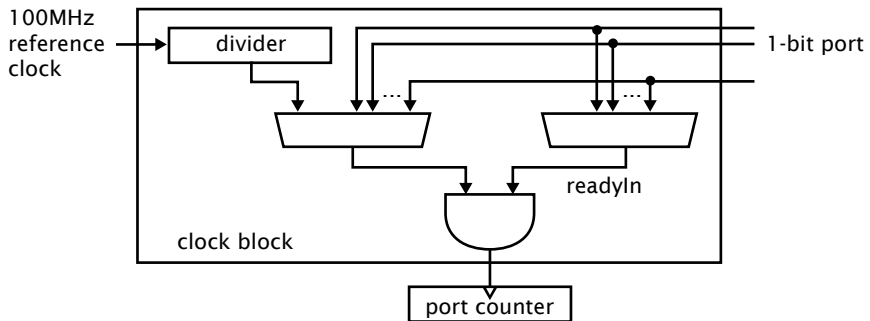

Figure 6:
Clock block
diagram

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
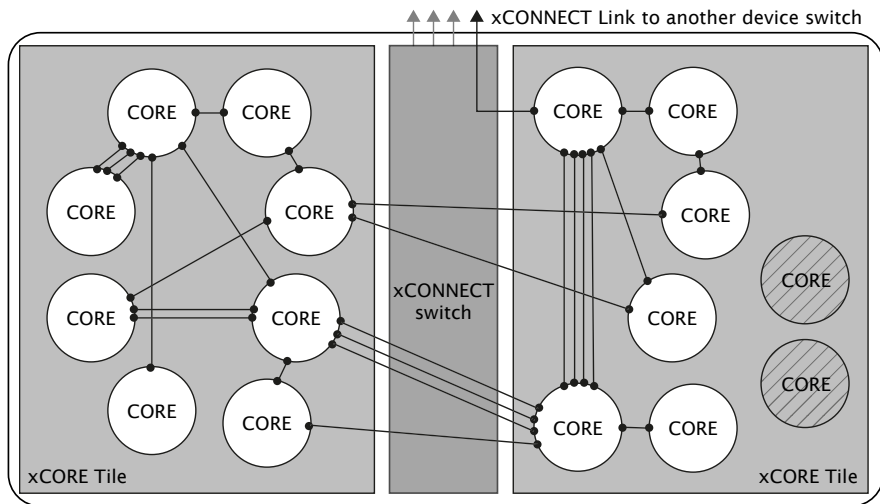
Figure 7:
Switch, links and channel ends

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 and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between tiles (up to 313 MBit/s), 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.

A package pin OSC_EXT_N is used to select the oscillator to use on boot. It must be grounded to select an external resonator or connected to VDDIO to select the on-chip 20 MHz oscillator. If an external resonator is used, then it must be in the range 5-100 MHz. If the USB PHY is used, then an external crystal (12 or 24 MHz) or an external oscillator (12, 24, 48, or 96 MHz) is required in order to provide a stable USB clock. Two more package pins, MODE0 and MODE1 are used to inform the node of the frequency.

The analog node runs at the frequency provided by the oscillator. Hence, increasing the clock frequency will speed up operation of the analog node, and will speed up communicating data with the digital node. The digital node has a PLL.

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock.

The PLL multiplication value is selected through the two MODE pins, and can be changed by software to speed up the tile or use less power. The MODE pins are set as shown in Figure 8:

Figure 8:
PLL multiplier values and MODE pins

Oscillator Frequency	MODE 1	0	Tile Frequency	PLL Ratio	PLL settings OD	F	R
5-13 MHz	0	0	130-399.75 MHz	30.75	1	122	0
13-20 MHz	1	1	260-400.00 MHz	20	2	119	0
20-48 MHz	1	0	167-400.00 MHz	8.33	2	49	0
48-100 MHz	0	1	196-400.00 MHz	4	2	23	0

Figure 8 also lists the values of OD, F and R, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

OD, F and R must be chosen so that $0 \leq R \leq 63$, $0 \leq F \leq 4095$, $0 \leq OD \leq 7$, and $260MHz \leq F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \leq 1.3GHz$. The OD, F, and R values can be modified by writing to the digital node PLL configuration register.

The MODE pins must be held at a static value during and after deassertion of the system reset.

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.
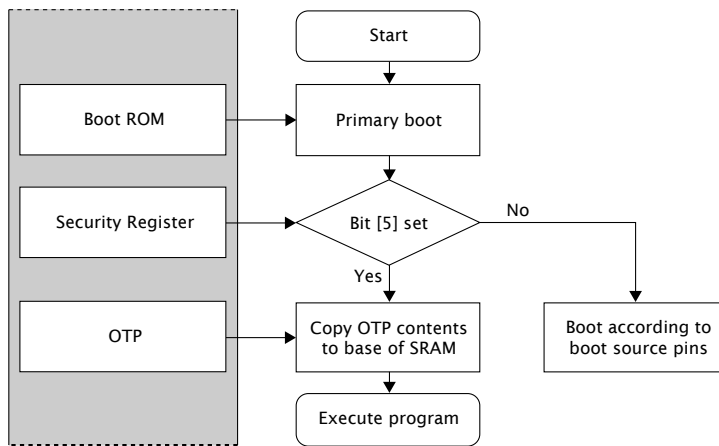

Figure 9:
Boot procedure

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.

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.

9.3 Boot from OTP

If an xCORE tile is set to use secure boot (see Figure 9), the boot image is read from address 0 of the OTP memory in the tile's security module.

This feature can be used to implement a secure bootloader which loads an encrypted image from external flash, decrypts and CRC checks it with the processor, and discontinues the boot process if the decryption or CRC check fails. XMOS provides a default secure bootloader that can be written to the OTP along with secret decryption keys.

Each tile has its own individual OTP memory, and hence some tiles can be booted from OTP while others are booted from SPI or the channel interface. This enables systems to be partially programmed, dedicating one or more tiles to perform a particular function, leaving the other tiles user-programmable.

9.4 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 12 provide a strong level of protection and are sufficient for providing strong IP security.

10 Memory

10.1 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.

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`.

10.2 SRAM

Each xCORE Tile integrates a single 64KB SRAM bank for both instructions and data. All internal memory is 32 bits wide, and instructions are either 16-bit or

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 processor is forced to boot from address 0 of the OTP, allowing the processor boot ROM to be bypassed (*see* §9).
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 12:
Security
register
features

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.

10.3 Deep Sleep Memory

The XS1-U8A-128-FB217 device includes 128 bytes of deep sleep memory for state storage during sleep mode. Deep sleep memory is volatile and if device input power is remove, the data will be lost.

11 USB PHY

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, libxud_s.a, is provided to implement USB device functionality.

11.1 Logical Core Requirements

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

To guarantee correct operation the USB logical core must run at at least 80 MIPS, and the logical cores that communicate with the USB core must also run at 80 MIPS. This means that no more than six logical cores execute at any one time on a 500MHz device.

12 Analog-to-Digital Converter

The device has a 12-bit 1MSample/second Successive Approximation Register (SAR) Analogue to Digital Converter (ADC). It has 8 input pins which are multiplexed into the ADC. The sampling of the ADC is controlled using GPIO pin X0D24 that is triggered either by writing to port 1I, or by driving the pin externally. On each rising edge of the sample pin the ADC samples, holds and converts the data value from one of the analog input pins. Each of the 8 inputs can be enabled individually. Each of the enabled analog inputs is sampled in turn, on successive rising edges of the sample pin. The data is transmitted to the channel-end that the user configures during initialization of the ADC. Data is transmitted over the channel in individual packets, or in packets that contain multiple consecutive samples. The ADC uses an external reference voltage, nominally 3V3, which represents the full range of the ADC. The ADC configuration registers are documented in Appendix G.

The minimum latency for reading a value from the ADC into the xCORE register is shown in Figure 13:

	Sample	Tile clock frequency	Start of packet	Subsequent samples
Figure 13: Minimum latency to read sample from ADC to xCORE	32-bit	500 MHz	840 ns	710 ns
	32-bit	400 MHz	870 ns	740 ns
	16-bit	500 MHz	770 ns	640 ns
	16-bit	400 MHz	800 ns	670 ns

13 Supervisor Logic

An independent supervisor circuit provides power-on-reset, brown-out, and watchdog capabilities. This facilitates the design of systems that fail gracefully, whilst keeping BOM costs down.

The reset supervisor holds the chip in reset until all power supplies are good. This provides a power-on-reset (POR). An external reset is optional and the pin RST_N can be left not-connected.

If at any time any of the power supplies drop because of too little supply or too high a demand, the power supervisor will bring the chip into reset until the power supplies have been restored. This will reboot the system as if a cold-start has happened.

The 16-bit watchdog timer provides 1ms accuracy and runs independently of the real-time counter. It can be programmed with a time-out of between 1 ms and 65 seconds (Appendix E). If the watchdog is not set before it times out, the XS1-U8A-128-FB217 is reset. On boot, the program can read a register to test whether the reset was due to the watchdog. The watchdog timer is only enabled and clocked whilst the processor is in the AWAKE power state.

14 Energy management

XS1-U8A-128-FB217 devices can be powered by:

▶ An external 5v core and 3.3v I/O supply, increasing efficiency for USB bus powered applications.

▶ A single 3.3v supply.

14.1 DC-DC

XS1-U8A-128-FB217 devices include two DC-DC buck converters which can be configured to take input voltages between 3.3-5V power supply and output circuit voltages (nominally 1.8V and 1.0V) required by the analog peripherals and digital node.

14.2 Power mode controller

The device transitions through multiple states during the power-up and powerdown process.

The device is quiescent in the ASLEEP state, and is running in the AWAKE state. The other states allow a controlled transition between AWAKE and ASLEEP.

A transition from AWAKE state to ASLEEP state is instigated by a sleep request: either a write to the general control register or from the USB block requesting entry to standby mode. Sleep requests must only be made in the AWAKE state.

A transition from the ASLEEP state into the AWAKE state is instigated by a wakeup request triggered by a request from the USB block to exit standby mode an input, or a timer. The device only responds to a wakeup stimulus in the ASLEEP state. If wakeup stimulus occurs whilst transitioning from AWAKE to ASLEEP, the appropriate response occurs when the ASLEEP state is reached.

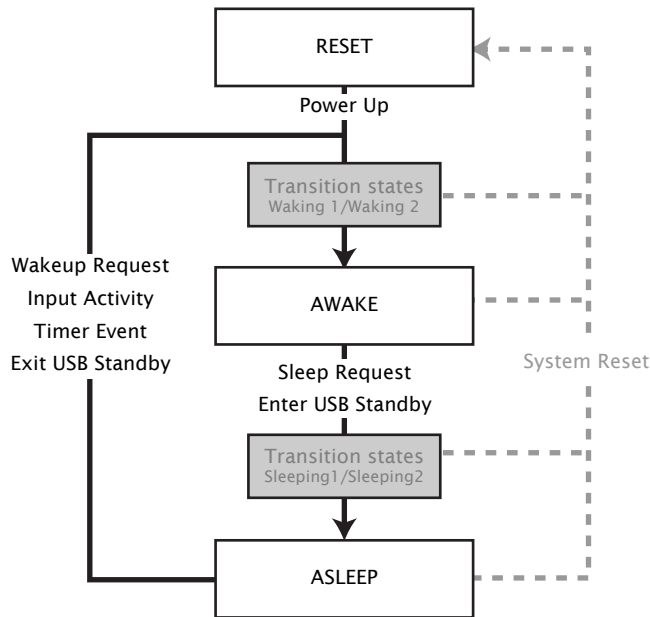Configuration is through a set of registers documented in Appendix K.

Figure 14:
XS1-U8A-128-
FB217 Power
Up States and
Transitions

14.3 Deep Sleep Modes and Real-Time Counter

The normal mode in which the XS1-U8A-128-FB217 operates is the AWAKE mode. In this mode, all cores, memory, and peripherals operate as normal. To save power, the XS1-U8A-128-FB217 can be put into a deep sleep mode, called ASLEEP, where the digital node is powered down, and most peripherals are powered down. The XS1-U8A-128-FB217 will stay in the ASLEEP mode until one of three conditions:

1. An external pin is asserted or deasserted (set by the program);

2. The 64-bit real-time counter reaches a value set by the program; or

3. The USB host (if USB is enabled) performs a wakeup.

When the chip is awake, the real-time counter counts the number of clock ticks on the oscillator. As such, the real-time counter will run at a fixed ratio, but synchronously with the 100 MHz timers on the xCORE Tile. When asleep, the real-time counter can be automatically switched to the 31,250 Hz silicon oscillator to save power (see Appendix I). To ensure that the real-time counter increases linearly over time, a programmable value is added to the counter on every 31,250 Hz clock-tick. This means that the clock will run at a granularity of 31,250 Hz but still maintain real-time in terms of the frequency of the main oscillator. If an accurate clock is required, even whilst asleep, then an external crystal or oscillator shall be provided that is used in both AWAKE and ASLEEP state.