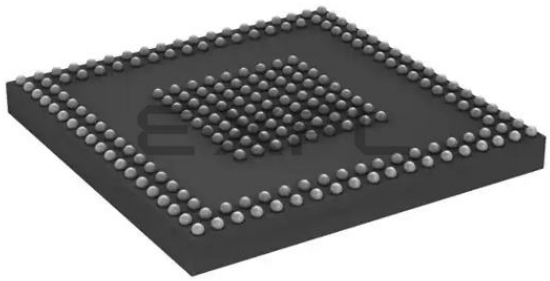

Welcome to E-XFL.COM

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 12-Core
Speed	1000MIPS
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	0°C ~ 70°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-a12a-128-fb217-c10

Signal	Function	Type	Properties

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 (90)			
Signal	**Function**	**Type**	**Properties**
X0D00	$1A^0$	I/O	PD_S, R_S
X0D01	XLA^4_{out} $1B^0$	I/O	PD_S, R_S
X0D02	XLA^3_{out} $4A^0$ $8A^0$ $16A^0$ $32A^{20}$	I/O	PD_S, R_U
X0D03	XLA^2_{out} $4A^1$ $8A^1$ $16A^1$ $32A^{21}$	I/O	PD_S, R_U
X0D04	XLA^1_{out} $4B^0$ $8A^2$ $16A^2$ $32A^{22}$	I/O	PD_S, R_U
X0D05	XLA^0_{out} $4B^1$ $8A^3$ $16A^3$ $32A^{23}$	I/O	PD_S, R_U
X0D06	XLA^0_{in} $4B^2$ $8A^4$ $16A^4$ $32A^{24}$	I/O	PD_S, R_U
X0D07	XLA^1_{in} $4B^3$ $8A^5$ $16A^5$ $32A^{25}$	I/O	PD_S, R_U
X0D08	XLA^2_{in} $4A^2$ $8A^6$ $16A^6$ $32A^{26}$	I/O	PD_S, R_U
X0D09	XLA^3_{in} $4A^3$ $8A^7$ $16A^7$ $32A^{27}$	I/O	PD_S, R_U
X0D10	XLA^4_{in} $1C^0$	I/O	PD_S, R_S
X0D11	$1D^0$	I/O	PD_S, R_S
X0D12	$1E^0$	I/O	PD_S, R_U
X0D13	XLB^4_{out} $1F^0$	I/O	PD_S, R_U
X0D14	XLB^3_{out} $4C^0$ $8B^0$ $16A^8$ $32A^{28}$	I/O	PD_S, R_U
X0D15	XLB^2_{out} $4C^1$ $8B^1$ $16A^9$ $32A^{29}$	I/O	PD_S, R_U
X0D16	XLB^1_{out} $4D^0$ $8B^2$ $16A^{10}$	I/O	PD_S, R_U

(continued)

5 Example Application Diagram

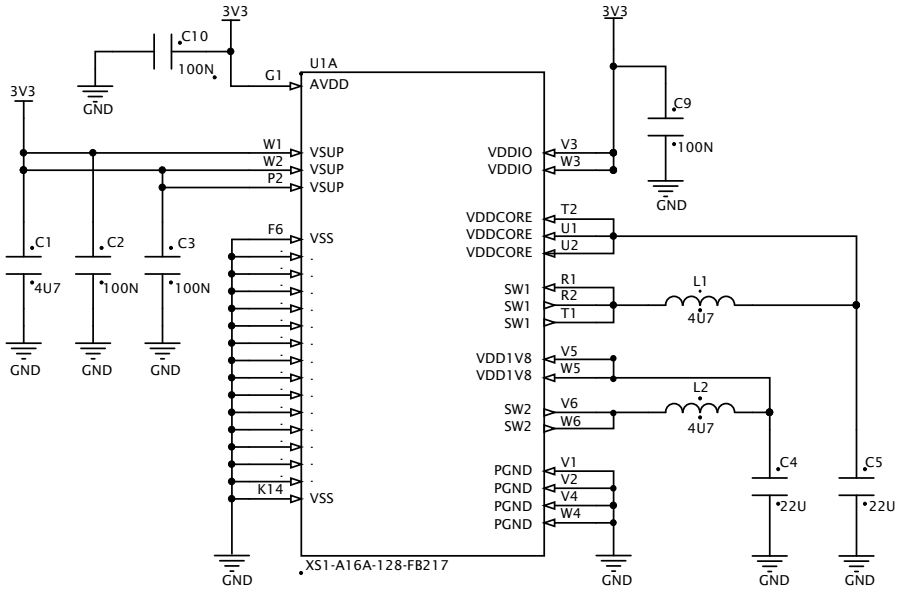

Figure 2: Simplified Reference Schematic

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 ADC and Power Management

Each XS1-A12A-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 6 active logical cores, which issue instructions down a shared four-stage pipeline. Instructions from the active cores are issued round-robin. If up to four logical cores are active, each core is allocated a quarter of the processing cycles. If more than four logical cores are active, each core is allocated at least $1/n$ cycles (for n cores). Figure 4 shows the guaranteed core performance depending on the number of cores used.

Figure 4: Logical core performance

Speed grade	MIPS	Frequency	Minimum MIPS per core (for *n* cores)							
			1	2	3	4	5	6		
8	800 MIPS	400 MHz	100	100	100	100	80	67		
10	1000 MIPS	500 MHz	125	125	125	125	100	83		

There is no way that the performance of a logical core can be reduced below these predicted levels. Because cores may be delayed on I/O, however, their unused processing cycles can be taken by other cores. This means that for more than four logical cores, the performance of each core is often higher than the predicted minimum but cannot be guaranteed.

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.

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.

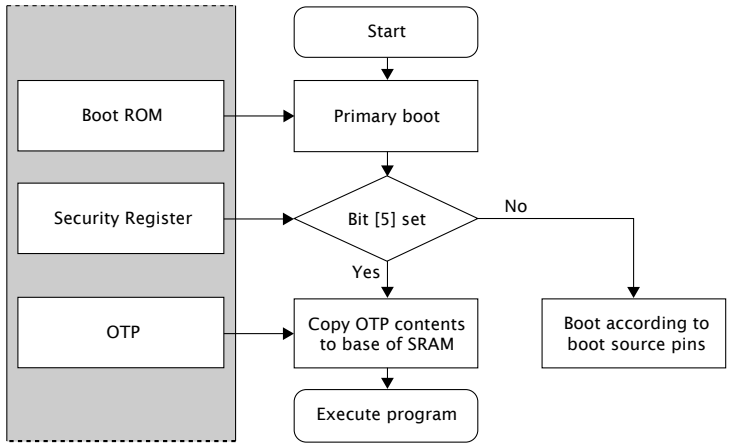

Figure 9: Boot procedure

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.

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 F.

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

Figure 13: Minimum latency to read sample from ADC to xCORE

Sample	Tile clock frequency	Start of packet	Subsequent samples
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

12 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-A12A-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.

13 Energy management

XS1-A12A-128-FB217 devices can be powered by:

- An external 5v core and 3.3v I/O supply.
- A single 3.3v supply.

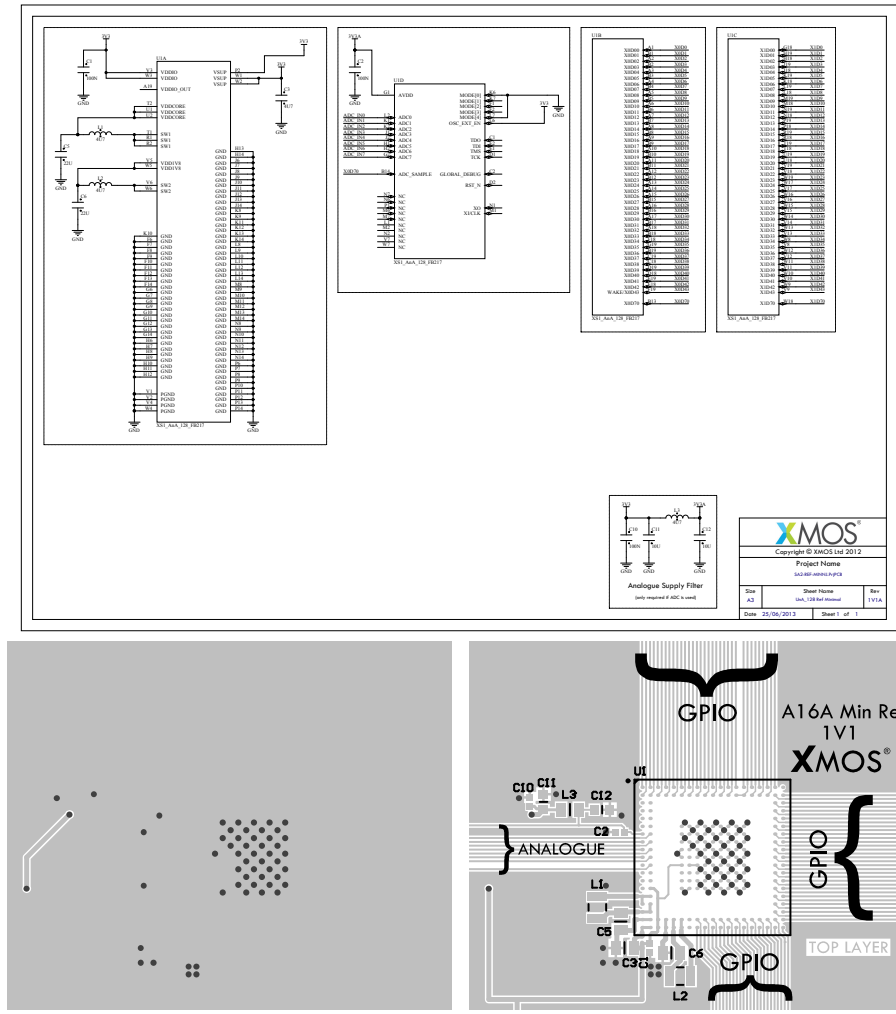

Figure 23: Example minimal system schematic, with top and bottom layout of a 2-layer PCB

17.5 Digital I/O Characteristics

Figure 28: 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

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

Figure 29: ESD stress voltage

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

17.7 Device Timing Characteristics

Figure 30: 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	

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	

B.19 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over JTAG. This is the same set of registers as the Debug Scratch registers in the xCORE tile configuration.

0x20 .. 0x27: Debug scratch

Bits	Perm	Init	Description
31:0	DRW		Value.

B.20 Instruction breakpoint address: 0x30 .. 0x33

This register contains the address of the instruction breakpoint. If the PC matches this address, then a debug interrupt will be taken. There are four instruction breakpoints that are controlled individually.

0x30 .. 0x33: Instruction breakpoint address

Bits	Perm	Init	Description
31:0	DRW		Value.

B.21 Instruction breakpoint control: 0x40 .. 0x43

This register controls which logical cores may take an instruction breakpoint, and under which condition.

0x40 .. 0x43: Instruction breakpoint control

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	Set to 1 to cause an instruction breakpoint if the PC is not equal to the breakpoint address. By default, the breakpoint is triggered when the PC is equal to the breakpoint address.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.22 Data watchpoint address 1: 0x50 .. 0x53

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

0x06: PLL settings

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:23	RW		OD: Output divider value The initial value depends on pins MODE0 and MODE1.
22:21	RO	-	Reserved
20:8	RW		F: Feedback multiplication ratio The initial value depends on pins MODE0 and MODE1.
7	RO	-	Reserved
6:0	RW		R: Oscilator input divider value The initial value depends on pins MODE0 and MODE1.

D.6 System switch clock divider: 0x07

Sets the ratio of the PLL clock and the switch clock.

0x07: System switch clock divider

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	Switch clock divider. The PLL clock will be divided by this value plus one to derive the switch clock.

D.7 Reference clock: 0x08

Sets the ratio of the PLL clock and the reference clock used by the node.

0x08: Reference clock

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	3	Architecture reference clock divider. The PLL clock will be divided by this value plus one to derive the 100 MHz reference clock.

D.8 Directions 0-7: 0x0C

This register contains eight directions, for packets with a mismatch in bits 7..0 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

E Analogue Node Configuration

The analogue 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).

Figure 44: Summary

Number	Perm	Description
0x00	RO	Device identification register
0x04	RW	Node configuration register
0x05	RW	Node identifier
0x50	RW	Reset and Mode Control
0x51	RW	System clock frequency
0x80	RW	Link Control and Status
0xD6	RW	1 KHz Watchdog Control
0xD7	RW	Watchdog Disable

E.1 Device identification register: 0x00

This register contains version information, and information on power-on behavior.

0x00: Device identification register

Bits	Perm	Init	Description
31:24	RO	0x0F	Chip identifier
23:17	RO	-	Reserved
16	RO	pin	Oscillator used on power-up. This is set by the OSC_EXT_N pin: 0: boot from crystal; 1: boot from on-silicon 20 MHz oscillator.
15:8	RO	0x02	Revision number of the analogue block
7:0	RO	0x00	Version number of the analogue block

E.2 Node configuration register: 0x04

This register is used to set the communication model to use (1 or 3 byte headers), and to prevent any further updates.

0x04: Node configuration register

Bits	Perm	Init	Description
31	RW	0	Set to 1 to disable further updates to the node configuration and link control and status registers.
30:1	RO	-	Reserved
0	RW	0	Header mode. 0: 3-byte headers; 1: 1-byte headers.

E.3 Node identifier: 0x05

0x05: Node identifier

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

E.4 Reset and Mode Control: 0x50

The XS1-S has two main reset signals: a system-reset and an xCORE Tile-reset. System-reset resets the whole system including external devices, whilst xCORE Tile-reset resets the xCORE Tile(s) only. The resets are induced either by software (by a write to the register below) or by one of the following:

* External reset on RST_N (System reset)

* Brown out on one of the power supplies (System reset)

* Watchdog timer (System reset)

* Sleep sequence (xCORE Tile reset)

* Clock source change (xCORE Tile reset)

The minimum system reset duration is achieved when the fastest permissible clock is used. The reset durations will be proportionately longer when a slower clock is used. Note that the minimum system reset duration allows for all power rails except the VOUT2 to turn off, and decay.

The length of the system reset comes from an internal counter, counting 524,288 oscillator clock cycles which gives the maximum time allowable for the supply rails to discharge. The system reset duration is a balance between leaving a long time for the supply rails to discharge, and a short time for the system to boot. Example reset times are 44 ms with a 12 MHz oscillator or 5.5 ms with a 96 MHz oscillator.

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

0x50: Reset and Mode Control

E.5 System clock frequency: 0x51

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	25	Oscillator clock frequency in MHz rounded up to the nearest integer value. Only values between 5 and 100 MHz are valid - writes outside this range are ignored and will be NACKed. This field must be set on start up of the device and any time that the input oscillator clock frequency is changed. It must contain the system clock frequency in MHz rounded up to the nearest integer value. The following functions depend on the correct frequency settings: * Processor reset delay * The watchdog clock * The real-time clock when running in sleep mode

0x51: System clock frequency

E.8 Watchdog Disable: 0xD7

To enable the watchdog, write 0 to this register. To disable the watchdog, write the value 0x0D1SAB1E to this register.

0xD7: Watchdog Disable

Bits	Perm	Init	Description
31:0	RW	0x0D15AB1E	A value of 0x0D15AB1E written to this register resets and disables the watchdog timer.

F ADC Configuration

The device has a 12-bit Analogue to Digital Converter (ADC). It has multiple input pins, and on each positive clock edge on port 1I, it samples and converts a value on the next input pin. The data is transmitted to a channel-end that must be set on enabling the ADC input pin.

The *ADC* is peripheral 2. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 2, ...)` and `read_periph_32(device, 2, ...)` for reads and writes).

Figure 45: Summary

Number	Perm	Description
0x00	RW	ADC Control input pin 0
0x04	RW	ADC Control input pin 1
0x08	RW	ADC Control input pin 2
0x0C	RW	ADC Control input pin 3
0x10	RW	ADC Control input pin 4
0x14	RW	ADC Control input pin 5
0x18	RW	ADC Control input pin 6
0x1C	RW	ADC Control input pin 7
0x20	RW	ADC General Control

F.1 ADC Control input pin 0: 0x00

Controls specific to ADC input pin 0.

I Real time clock Configuration

The *Real time clock* is peripheral 5. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 5, ...)` and `read_periph_32(device, 5, ...)` for reads and writes).

Figure 48: Summary

Number	Perm	Description
0x00	RW	Real time counter least significant 32 bits
0x04	RW	Real time counter most significant 32 bits

I.1 Real time counter least significant 32 bits: 0x00

This registers contains the lower 32-bits of the real-time counter.

0x00: Real time counter least significant 32 bits

Bits	Perm	Init	Description
31:0	RO	0	Least significant 32 bits of real-time counter.

I.2 Real time counter most significant 32 bits: 0x04

This registers contains the upper 32-bits of the real-time counter.

0x04: Real time counter most significant 32 bits

Bits	Perm	Init	Description
31:0	RO	0	Most significant 32 bits of real-time counter.

J Power control block Configuration

The *Power control block* is peripheral 6. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 6, ...)` and `read_periph_32(device, 6, ...)` for reads and writes).

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

Figure 49: Summary

J.1 General control: 0x00

This register controls the basic settings for power modes.

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.

- TDO to pin 13 of the xSYS header
- RST_N to pin 15 of the xSYS header
- If MODE2 is configured high, connect MODE2 to pin 3 of the xSYS header. Do not connect to VDDIO.
- If MODE3 is configured high, connect MODE3 to pin 3 of the xSYS header. Do not connect to VDDIO.

The RST_N net should be open-drain, active-low, and have a pull-up to VDDIO.

M.3 Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section M.2, and then connect a 2-wire xCONNECT Link to the xSYS header. The links can be found in the Signal description table (Section 4): they are labelled XLA, XLB, etc in the function column. The 2-wire link comprises two inputs and outputs, labelled ${}^{1}_{out}$, ${}^{0}_{out}$, ${}^{0}_{in}$, and ${}^{1}_{in}$. For example, if you choose to use XLB of tile 0 for xSCOPE I/O, you need to connect up XLB^{1}_{out}, XLB^{0}_{out}, XLB^{0}_{in}, XLB^{1}_{in} as follows:

- XLB^{1}_{out} (X0D16) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- XLB^{0}_{out} (X0D17) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- XLB^{0}_{in} (X0D18) to pin 14 of the xSYS header.
- XLB^{1}_{in} (X0D19) to pin 18 of the xSYS header.

N Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XS1-A12A-128-FB217. Each of the following sections contains items to check for each design.

N.1 Clock

☐ Pins MODE0 and MODE1 are set to the correct value for the chosen frequency. The MODE settings are shown in the Oscillator section, Section 8. If you have a choice between two values, choose the value with the highest multiplier ratio since that will boot faster.

☐ OSC_EXT_N is tied to ground (for use with a crystal or oscillator) or tied to VDDIO (for use with the internal oscillator). If using the internal oscillator, set MODE0 and MODE1 to be for the 20-48 MHz range (Section 8).

☐ If you have used an oscillator, it is a 1V8 oscillator. (Section 16)

N.2 USB ULPI Mode

This section can be skipped if you do not have an external USB PHY.

☐ If using ULPI, the ULPI signals are connected to specific ports as shown in Section K.

☐ If using ULPI, the ports that are used internally are not connected, see Section K. (Note that this limitation only applies when the ULPI is enabled, they can still be used before or after the ULPI is being used.)

N.3 Boot

☐ The device is connected to a SPI flash for booting, connected to X0D0, X0D01, X0D10, and X0D11 (Section 9). If not, you must boot the device through OTP or JTAG.

☐ The device that is connected to flash has both MODE2 and MODE3 connected to pin 3 on the xSYS Header (MSEL). If no debug adapter connection is supported (not recommended) MODE2 and MODE3 are to be left NC (Section 9). MODE4 is set in accordance with Section 9.

☐ The SPI flash that you have chosen is supported by **xflash**, or you have created a specification file for it.

N.4 JTAG, XScope, and debugging

- ☐ You have decided as to whether you need an XSYS header or not (Section M)
- ☐ If you included an XSYS header, you connected pin 3 to any MODE2/MODE3 pin that would otherwise be NC (Section M).
- ☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section M).

N.5 GPIO

- ☐ You have not mapped both inputs and outputs to the same multi-bit port.

N.6 Multi device designs

Skip this section if your design only includes a single XMOS device.

- ☐ One device is connected to a SPI flash for booting.
- ☐ Devices that boot from link have MODE2 grounded and MODE3 NC. These device must have link XLB connected to a device to boot from (see 9).
- ☐ If you included an XSYS header, you have included buffers for RST_N, TMS, TCK, MODE2, and MODE3 (Section L).

R Revision History

Date	Description
2013-04-16	First release
2013-07-19	Updated Features list with available ports and links - Section 2 Simplified link bits in Signal Description - Section 4 New JTAG, xSCOPE and Debugging appendix - Section M New Schematics Design Check List - Section N New PCB Layout Design Check List - Section O
2013-12-09	Added Industrial Ambient Temperature - Section 17.1 Annotated V(ACC) parameter - Section 17.2 Updated V(IH) parameter - Section 17.9 Updated V(OH) parameter - Section 17.5
2014-02-26	Added C8 and I8 parts - Section 19
2014-03-25	Added footnotes to DC and Switching Characteristics - Section 17
2015-04-14	Updated Introduction - Section 1; Pin Configuration - Section 3; Signal Description - Section 4