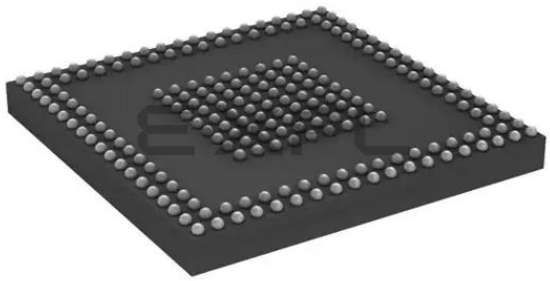

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	XCore
Core Size	32-Bit 10-Core
Speed	1000MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	73
Program Memory Size	128KB (32K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	3V ~ 3.6V
Data Converters	A/D 8x12b
Oscillator Type	Internal
Operating Temperature	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-u10a-128-fb217-c10

Signal	Function						Type	Properties
X0D21	XLB^3_{in}		$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	PD_S
X0D22	XLB^4_{in}	$1G^0$					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^4_{out}	$1B^0$					I/O	PD_S, R_S
X1D02	XLA^3_{out}		$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	PD_S
X1D03	XLA^2_{out}		$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	I/O	PD_S
X1D04	XLA^1_{out}		$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O	PD_S
X1D05	XLA^0_{out}		$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O	PD_S
X1D06	XLA^0_{in}		$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O	PD_S
X1D07	XLA^1_{in}		$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O	PD_S
X1D08	XLA^2_{in}		$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	PD_S
X1D09	XLA^3_{in}		$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	PD_S
X1D10	XLA^4_{in}	$1C^0$					I/O	PD_S, R_S
X1D11		$1D^0$					I/O	PD_S, R_S
X1D12		$1E^0$					I/O	PD_S
X1D13	XLB^4_{out}	$1F^0$					I/O	PD_S
X1D14	XLB^3_{out}		$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	PD_S
X1D15	XLB^2_{out}		$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	PD_S
X1D16	XLB^1_{out}		$4D^0$	$8B^2$	$16A^{10}$		I/O	PD_S
X1D17	XLB^0_{out}		$4D^1$	$8B^3$	$16A^{11}$		I/O	PD_S
X1D18	XLB^0_{in}		$4D^2$	$8B^4$	$16A^{12}$		I/O	PD_S
X1D19	XLB^1_{in}		$4D^3$	$8B^5$	$16A^{13}$		I/O	PD_S
X1D20	XLB^2_{in}		$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	PD_S
X1D21	XLB^3_{in}		$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	PD_S
X1D22	XLB^4_{in}	$1G^0$					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^4_{out}					$32A^0$	I/O	PD_S
X1D50	XLC^3_{out}					$32A^1$	I/O	PD_S

(continued)

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

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-U10A-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 5 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			
10	1000 MIPS	500 MHz	125	125	125	125	100			

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.

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.

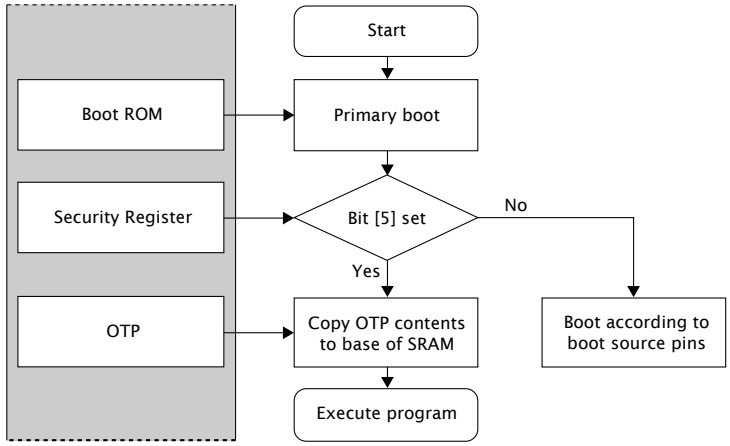

Figure 9: Boot procedure

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.

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

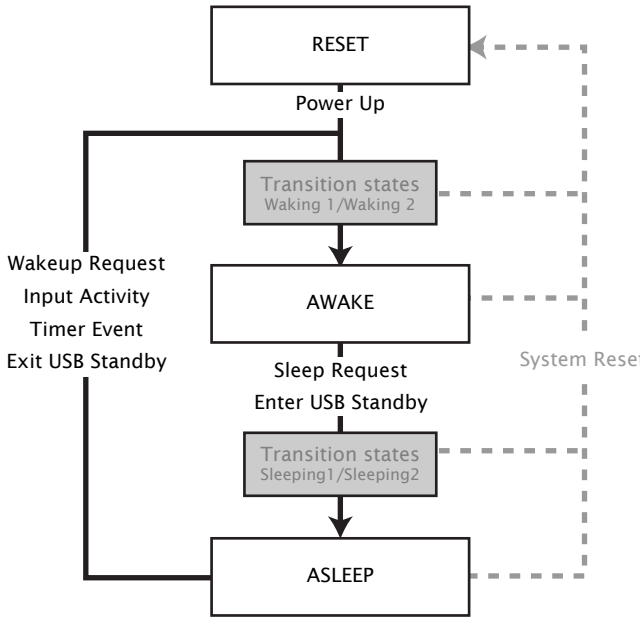

Figure 14: XS1-U10A-128-FB217 Power Up States and Transitions

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.

14.3 Deep Sleep Modes and Real-Time Counter

The normal mode in which the XS1-U10A-128-FB217 operates is the AWAKE mode. In this mode, all cores, memory, and peripherals operate as normal. To save power, the XS1-U10A-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-U10A-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

15 JTAG

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

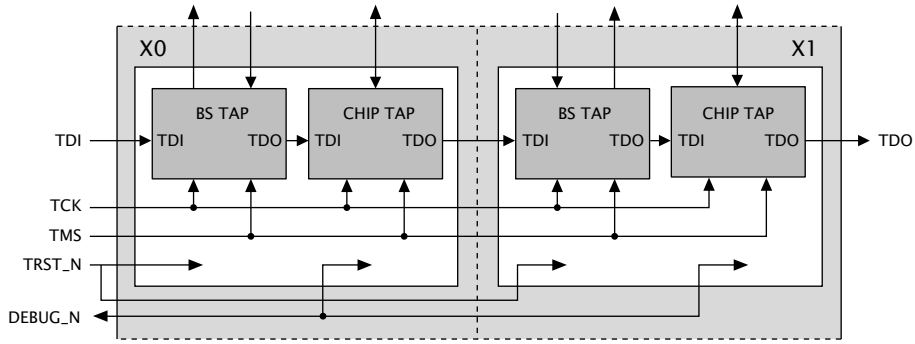

Figure 16: JTAG chain structure

The JTAG chain structure is illustrated in Figure 16. Directly after reset, three TAP controllers are present in the JTAG chain for each xCORE Tile: the debug TAP, the boundary scan TAP and the processor TAP. The debug TAP provides access into the peripherals including the ADC and USB. The boundary scan TAP is a standard 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. The processor TAP provides access into the xCORE Tile, switch and OTP for loading code and debugging.

The JTAG module can be reset by holding TMS high for five clock cycles.

The DEBUG_N pin is used to synchronize the debugging of multiple processors. This pin can operate in both output and input mode. In output mode and when configured to do so, DEBUG_N is driven low by the device when the processor hits a debug break point. Prior to this point the pin will be tri-stated. In input mode and when configured to do so, driving this pin low will put the processor into debug mode. Software can set the behavior of the processor based on this pin. This pin should have an external pull up of 4K7-47KΩ or left not connected in single core applications.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 17.

Figure 17: IDCODE return value

Bit31				Device Identification Register																								Bit0
Version				Part Number																Manufacturer Identity								1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	0	1	1	1
0				0				0				0				3				6				3				3

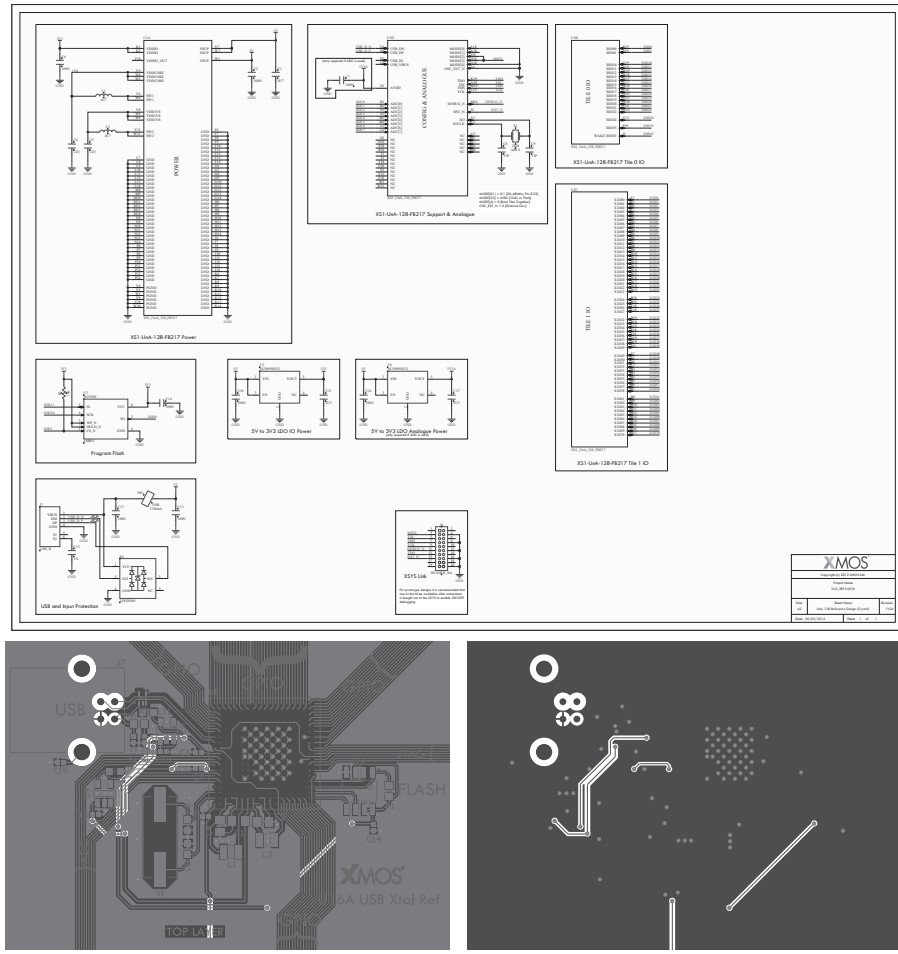

Figure 23: Example XTAL schematic, with top and bottom layout of a 2-layer PCB

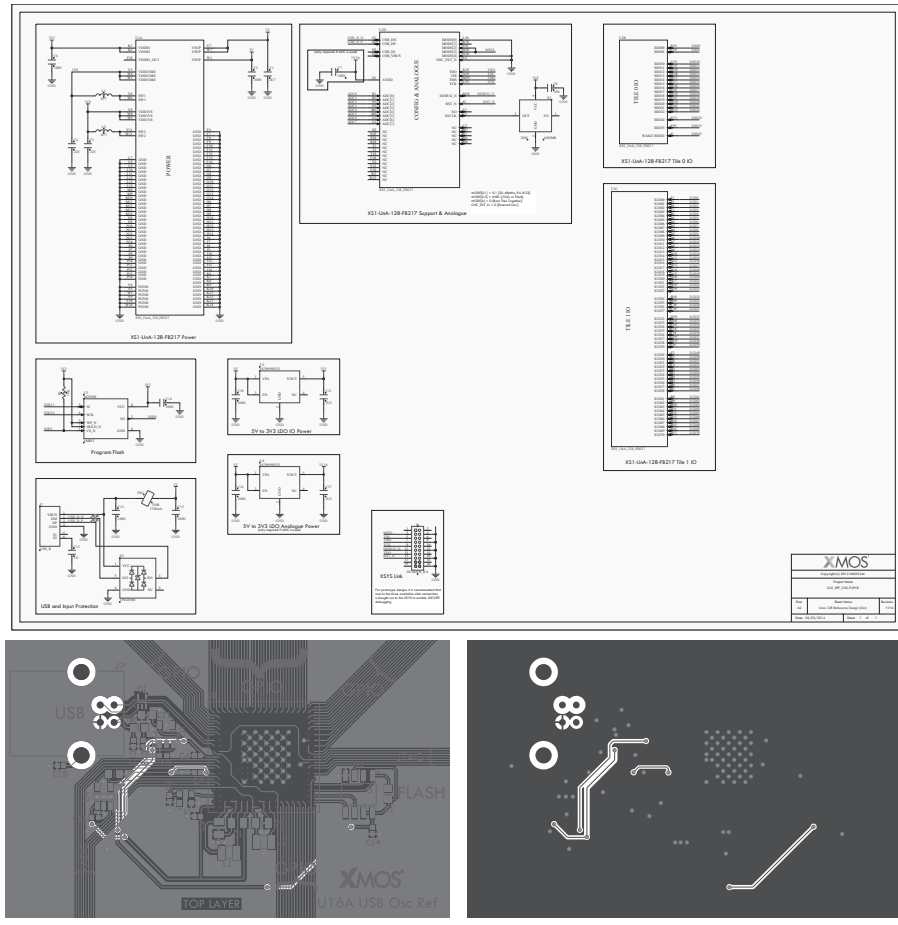

Figure 24: Example Oscillator schematic, with top and bottom layout of a 2-layer PCB

18.14 xConnect Link Performance

Figure 39: Link performance

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
B(2blinkP)	2b link bandwidth (packetized)			103	MBit/s	A, B
B(5blinkP)	5b link bandwidth (packetized)			271	MBit/s	A, B
B(2blinkS)	2b link bandwidth (streaming)			125	MBit/s	B
B(5blinkS)	5b link bandwidth (streaming)			313	MBit/s	B

A Assumes 32-byte packet in 3-byte header mode. Actual performance depends on size of the header and payload.

B 7.5 ns symbol time.

The asynchronous nature of links means that the relative phasing of CLK clocks is not important in a multi-clock system, providing each meets the required stability criteria.

18.15 JTAG Timing

Figure 40: JTAG timing

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			TBC	MHz	
f(TCK_B)	TCK frequency (boundary scan)			TBC	MHz	
T(SETUP)	TDO to TCK setup time	TBC			ns	A
T(HOLD)	TDO to TCK hold time	TBC			ns	A
T(DELAY)	TCK to output delay			TBC	ns	B

A Timing applies to TMS and TDI inputs.

B Timing applies to TDO output from negative edge of TCK.

All JTAG operations are synchronous to TCK.

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.

0x50 .. 0x53: Data watchpoint address 1

Bits	Perm	Init	Description
31:0	DRW		Value.

B.23 Data watchpoint address 2: 0x60 .. 0x63

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

0x60 .. 0x63: Data watchpoint address 2

Bits	Perm	Init	Description
31:0	DRW		Value.

B.24 Data breakpoint control register: 0x70 .. 0x73

This set of registers controls each of the four data watchpoints.

0x70 .. 0x73: Data breakpoint control register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:3	RO	-	Reserved
2	DRW	0	Set to 1 to enable breakpoints to be triggered on loads. Breakpoints always trigger on stores.
1	DRW	0	By default, data watchpoints trigger if memory in the range [Address1..Address2] is accessed (the range is inclusive of Address1 and Address2). If set to 1, data watchpoints trigger if memory outside the range (Address2..Address1) is accessed (the range is exclusive of Address2 and Address1).
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.25 Resources breakpoint mask: 0x80 .. 0x83

This set of registers contains the mask for the four resource watchpoints.

D Digital Node Configuration

The digital node control registers can be accessed using configuration reads and writes (use `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ...)` for reads and writes).

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

Figure 46: Summary

D.1 Device identification: 0x00

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

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

0x00: Device identification

D.2 System switch description: 0x01

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

0x01: System switch description

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Number of links on the switch.
15:8	RO		Number of cores that are connected to this switch.
7:0	RO		Number of links per processor.

D.3 Switch configuration: 0x04

This register enables the setting of two security modes (that disable updates to the PLL or any other registers) and the header-mode.

0x04: Switch configuration

Bits	Perm	Init	Description
31	RO	0	Set to 1 to disable any write access to the configuration registers in this switch.
30:9	RO	-	Reserved
8	RO	0	Set to 1 to disable updates to the PLL configuration register.
7:1	RO	-	Reserved
0	RO	0	Header mode. Set to 1 to enable 1-byte headers. This must be performed on all nodes in the system.

D.4 Switch node identifier: 0x05

This register contains the node identifier.

0x05: Switch node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	The unique 16-bit ID of this node. This ID is matched most-significant-bit first with incoming messages for routing purposes.

D.5 PLL settings: 0x06

An on-chip PLL multiplies the input clock up to a higher frequency clock, used to clock the I/O, processor, and switch, see Oscillator. Note: a write to this register will cause the tile to be reset.

Number	Perm	Description
0x00	WO	UIFM reset
0x04	RW	UIFM IFM control
0x08	RW	UIFM Device Address
0x0C	RW	UIFM functional control
0x10	RW	UIFM on-the-go control
0x14	RO	UIFM on-the-go flags
0x18	RW	UIFM Serial Control
0x1C	RW	UIFM signal flags
0x20	RW	UIFM Sticky flags
0x24	RW	UIFM port masks
0x28	RW	UIFM SOF value
0x2C	RO	UIFM PID
0x30	RO	UIFM Endpoint
0x34	RW	UIFM Endpoint match
0x38	RW	UIFM power signalling
0x3C	RW	UIFM PHY control

Figure 48: Summary

F.1 UIFM reset: 0x00

A write to this register with any data resets all UIFM state, but does not otherwise affect the phy.

0x00: UIFM reset

Bits	Perm	Init	Description
31:0	WO		Value.

F.2 UIFM IFM control: 0x04

General settings of the UIFM IFM state machine.

0x20: UIFM Sticky flags

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

F.10 UIFM port masks: 0x24

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

0x24: UIFM port masks

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

F.11 UIFM SOF value: 0x28

USB Start-Of-Frame counter

0x28: UIFM SOF value

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

F.12 UIFM PID: 0x2C

The last USB packet identifier received

J 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 52: Summary

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

J.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.

J.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.

K 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).

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	1	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).

0x1C: Power supply states whilst SLEEPING1

K.9 Power supply states whilst SLEEPING2: 0x20

This register controls what state the power control block should be in when in the SLEEPING2 state. It also defines the time that the system shall stay in this state.

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).

K.11 DCDC control: 0x2C

This register controls the two DC-DC converters.

- ☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section M).

N.4 GPIO

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

N.5 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).