

Welcome to [E-XFL.COM](https://www.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	Obsolete
Core Processor	XCore
Core Size	32-Bit 12-Core
Speed	2000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	81
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	128-TQFP Exposed Pad
Supplier Device Package	128-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xuf212-512-tq128-i20

2 XUF212-512-TQ128 Features

► Multicore Microcontroller with Advanced Multi-Core RISC Architecture

- 12 real-time logical cores on 2 xCORE tiles
- Cores share up to 1000 MIPS
 - Up to 2000 MIPS in dual issue mode
- Each logical core has:
 - Guaranteed throughput of between $\frac{1}{5}$ and $\frac{1}{6}$ of tile MIPS
 - 16x32bit dedicated registers
- 167 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

► Programmable I/O

- 81 general-purpose I/O pins, configurable as input or output
 - Up to 25 x 1bit port, 12 x 4bit port, 8 x 8bit port, 4 x 16bit port
 - 4 xCONNECT links
- Port sampling rates of up to 60 MHz with respect to an external clock
- 64 channel ends (32 per tile) for communication with other cores, on or off-chip

► Memory

- 512KB internal single-cycle SRAM (max 256KB per tile) for code and data storage
- 16KB internal OTP (max 8KB per tile) for application boot code
- 2MB internal flash for application code and overlays

► 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

- 20: 1000 MIPS

► Power Consumption

- 570 mA (typical)

► 128-pin TQFP package 0.4 mm pitch

4 Signal Description

This section lists the signals and I/O pins available on the XUF212-512-TQ128. 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 has a weak pull-down or pull-up resistor. The resistor is enabled during and after reset. Enabling a link or port that uses the pin disables the resistor. Thereafter, the resistor can be enabled or disabled under software control. The resistor is designed to ensure defined logic input state for unconnected pins. It should not be used to pull external circuitry. Note that the resistors are highly non-linear and only a maximum pull current is specified in Section 13.2.
- ST: The IO pin has a Schmitt Trigger on its input.
- IOL/IOT/IOR: The IO pin is powered from VDDIOL, VDDIOT, and VDDIOR respectively

Power pins (10)			
Signal	Function	Type	Properties
GND	Digital ground	GND	
OTP_VCC	OTP power supply	PWR	
PLL_AGND	Analog ground for PLL	PWR	
PLL_AVDD	Analog PLL power	PWR	
USB_VDD	Digital tile power	PWR	
USB_VDD33	USB Analog power	PWR	
VDD	Digital tile power	PWR	
VDDIOL	Digital I/O power (left)	PWR	
VDDIOR	Digital I/O power (right)	PWR	
VDDIOT	Digital I/O power (top)	PWR	

JTAG pins (6)			
Signal	Function	Type	Properties
RST_N	Global reset input	Input	IOL, PU, ST
TCK	Test clock	Input	IOL, PD, ST
TDI	Test data input	Input	IOL, PU
TDO	Test data output	Output	IOL, PD
TMS	Test mode select	Input	IOL, PU
TRST_N	Test reset input	Input	IOL, PU, ST

Signal	Function	Type	Properties
X0D41	$X_0L0_{in}^{0}$ 8D ⁵ 16B ¹³	I/O	IOL, PD
X0D42	$X_0L0_{out}^{0}$ 8D ⁶ 16B ¹⁴	I/O	IOL, PD
X0D43	$X_0L0_{out}^{0}$ 8D ⁷ 16B ¹⁵	I/O	IOL, PD
X1D00	$X_0L7_{in}^{0}$ 1A ⁰	I/O	IOR, PD
X1D01	$X_0L7_{in}^{1}$ 1B ⁰	I/O	IOR, PD
X1D02	$X_0L4_{in}^{0}$ 4A ⁰ 8A ⁰ 16A ⁰ 32A ²⁰	I/O	IOR, PD
X1D03	$X_0L4_{out}^{1}$ 4A ¹ 8A ¹ 16A ¹ 32A ²¹	I/O	IOR, PD
X1D04	$X_0L4_{out}^{2}$ 4B ⁰ 8A ² 16A ² 32A ²²	I/O	IOR, PD
X1D05	$X_0L4_{out}^{2}$ 4B ¹ 8A ³ 16A ³ 32A ²³	I/O	IOR, PD
X1D06	$X_0L4_{out}^{3}$ 4B ² 8A ⁴ 16A ⁴ 32A ²⁴	I/O	IOR, PD
X1D07	$X_0L4_{out}^{4}$ 4B ³ 8A ⁵ 16A ⁵ 32A ²⁵	I/O	IOR, PD
X1D08	$X_0L7_{in}^{4}$ 4A ² 8A ⁶ 16A ⁶ 32A ²⁶	I/O	IOR, PD
X1D09	$X_0L7_{in}^{3}$ 4A ³ 8A ⁷ 16A ⁷ 32A ²⁷	I/O	IOR, PD
X1D10	1C ⁰	I/O	IOT, PD
X1D11	1D ⁰	I/O	IOT, PD
X1D14	4C ⁰ 8B ⁰ 16A ⁸ 32A ²⁸	I/O	IOR, PD
X1D15	4C ¹ 8B ¹ 16A ⁹ 32A ²⁹	I/O	IOR, PD
X1D16	$X_0L3_{in}^{1}$ 4D ⁰ 8B ² 16A ¹⁰	I/O	IOL, PD
X1D17	$X_0L3_{in}^{0}$ 4D ¹ 8B ³ 16A ¹¹	I/O	IOL, PD
X1D18	$X_0L3_{out}^{0}$ 4D ² 8B ⁴ 16A ¹²	I/O	IOL, PD
X1D19	$X_0L3_{out}^{1}$ 4D ³ 8B ⁵ 16A ¹³	I/O	IOL, PD
X1D20	4C ² 8B ⁶ 16A ¹⁴ 32A ³⁰	I/O	IOR, PD
X1D21	4C ³ 8B ⁷ 16A ¹⁵ 32A ³¹	I/O	IOR, PD
X1D26	4E ⁰ 8C ⁰ 16B ⁰	I/O	IOT, PD
X1D27	4E ¹ 8C ¹ 16B ¹	I/O	IOT, PD
X1D28	4F ⁰ 8C ² 16B ²	I/O	IOT, PD
X1D29	4F ¹ 8C ³ 16B ³	I/O	IOT, PD
X1D30	4F ² 8C ⁴ 16B ⁴	I/O	IOT, PD
X1D31	4F ³ 8C ⁵ 16B ⁵	I/O	IOT, PD
X1D32	4E ² 8C ⁶ 16B ⁶	I/O	IOT, PD
X1D33	4E ³ 8C ⁷ 16B ⁷	I/O	IOT, PD
X1D35	1L ⁰	I/O	IOL, PD
X1D36	1M ⁰ 8D ⁰ 16B ⁸	I/O	IOL, PD
X1D37	1N ⁰ 8D ¹ 16B ⁹	I/O	IOL, PD
X1D38	1O ⁰ 8D ² 16B ¹⁰	I/O	IOL, PD
X1D39	1P ⁰ 8D ³ 16B ¹¹	I/O	IOL, PD
X1D40	8D ⁴ 16B ¹²	I/O	IOT, PD
X1D41	8D ⁵ 16B ¹³	I/O	IOT, PD
X1D42	8D ⁶ 16B ¹⁴	I/O	IOT, PD
X1D43	8D ⁷ 16B ¹⁵	I/O	IOT, PD

usb pins (5)			
Signal	Function	Type	Properties
USB_DM	USB Serial Data Inverted	I/O	
USB_DP	USB Serial Data	I/O	
USB_ID	USB Device ID (OTG) - Reserved	I/O	
USB_RTUNE	USB resistor	I/O	
USB_VBUS	USB Power Detect Pin	I/O	

System pins (1)			
Signal	Function	Type	Properties
CLK	PLL reference clock	Input	IOL, PD, ST

5 Example Application Diagram

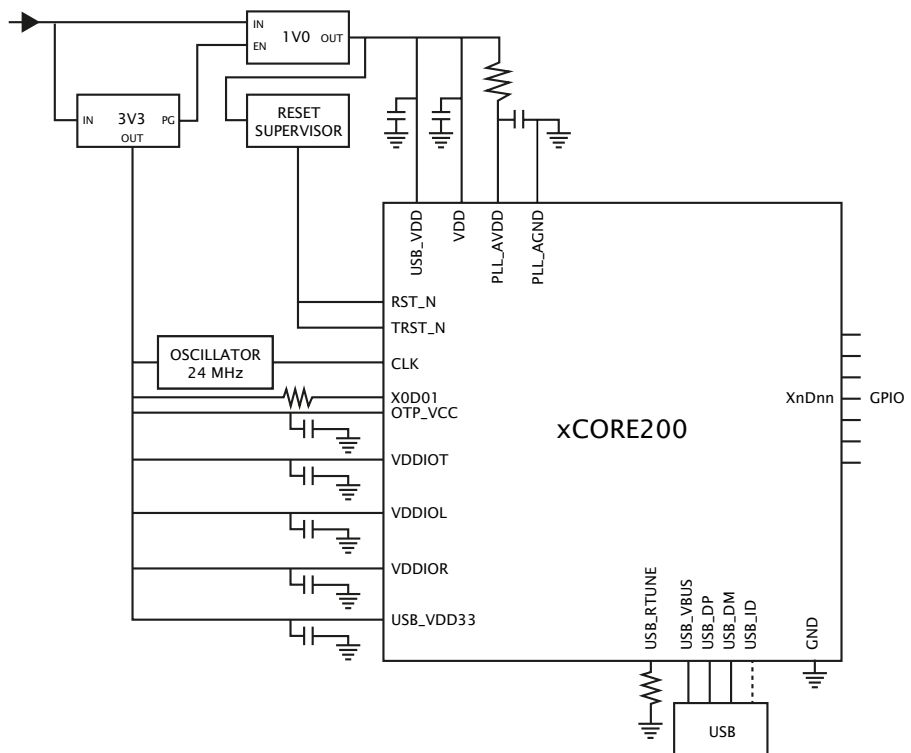


Figure 2:
Simplified
Reference
Schematic

- ▶ see Section 10 for details on the USB PHY
- ▶ see Section 12 for details on the power supplies and PCB design

6 Product Overview

The XUF212-512-TQ128 is a powerful device that consists of two xCORE Tiles, each comprising a flexible logical processing cores with tightly integrated I/O and on-chip memory.

6.1 Logical cores

Each tile has 6 active logical cores, which issue instructions down a shared five-stage pipeline. Instructions from the active cores are issued round-robin. If up to five logical cores are active, each core is allocated a fifth of the processing cycles. If more than five logical cores are active, each core is allocated at least $1/n$ cycles (for n cores). Figure 3 shows the guaranteed core performance depending on the number of cores used.

Figure 3:
Logical core
performance

Speed grade	MIPS	Frequency	Minimum MIPS per core (for n cores)					
			1	2	3	4	5	6
10	1000 MIPS	500 MHz	100	100	100	100	100	83

There is no way that the performance of a logical core can be reduced below these predicted levels (unless *priority threads* are used: in this case the guaranteed minimum performance is computed based on the number of priority threads as defined in the architecture manual). 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 five 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.

6.2 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.

6.3 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 XUF212-512-TQ128, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

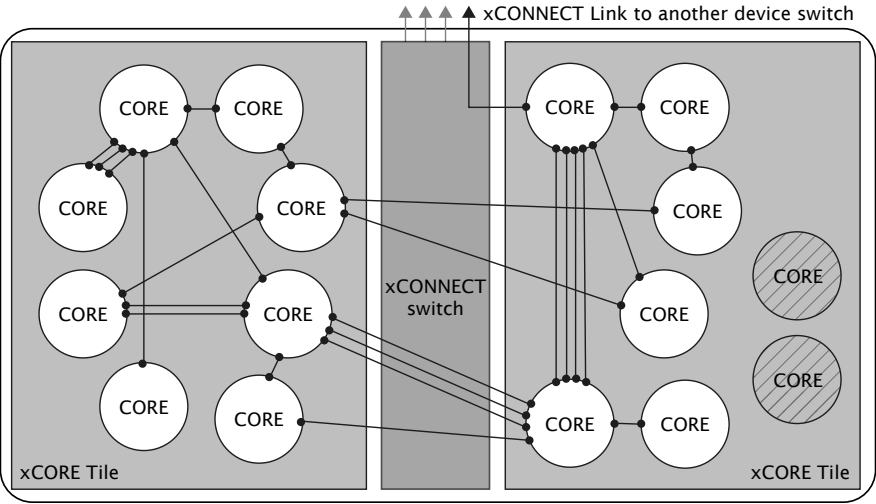


Figure 6:
Switch, links
and channel
ends

and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between xCORE Tiles (up to 250 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-UF Link Performance and Design Guide, [X2999](#).

7 PLL

The PLL creates a high-speed clock that is used for the switch, tile, and reference clock. The initial PLL multiplication value is shown in Figure 7:

Figure 7:
The initial PLL
multiplier
values

Oscillator Frequency	Tile Boot Frequency	PLL Ratio	PLL settings		
			OD	F	R
9-25 MHz	144-400 MHz	16	1	63	0

Figure 7 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.

If the USB PHY is used, then either a 24 MHz or 12 MHz oscillator must be used.

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 xCORE-200 Clock Frequency Control document.

8 Boot Procedure

The device is kept in reset by driving RST_N low. When in reset, all GPIO pins have a pull-down enabled. The processor must be held in reset until VDDIOL is in spec for at least 1 ms. When the device is taken out of reset by releasing RST_N the processor starts its internal reset process. After 15-150 μ s (depending on the input clock) the processor boots.

The device boots from a QSPI flash (IS25LQ016B) that is embedded in the device. The QSPI flash is connected to the ports on Tile 0 as shown in Figure 8. An external 1K resistor must connect X0D01 to VDDIOL. X0D10 should ideally not be connected. If X0D10 is connected, then a 150 ohm series resistor close to the device is recommended. X0D04..X0D07 should be not connected.

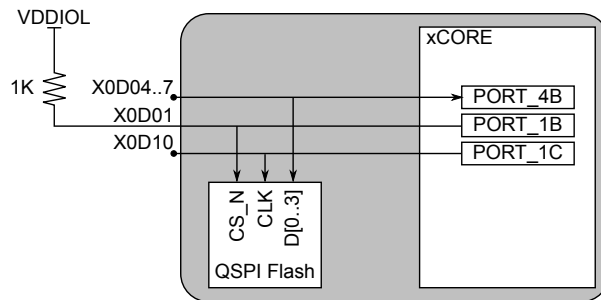


Figure 8:
QSPI port
connectivity

The xCORE Tile boot procedure is illustrated in Figure 9. If bit 5 of the security register (see §9.1) is set, the device boots from OTP. Otherwise, the device boots from the internal flash.

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

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 xCORE Tile is forced to boot from address 0 of the OTP, allowing the xCORE Tile boot ROM to be bypassed (see §8).
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.
	21..15	General purpose software accessible security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 10:
Security
register
features

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 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, XUD, is provided to implement *USB-device* functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 11. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles. Two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.

In any case, extra components (such as a ferrite bead and diodes) may be required for EMC compliance and ESD protection. Different wiring is required for USB-host and USB-OTG.

10.2 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.

11 JTAG

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

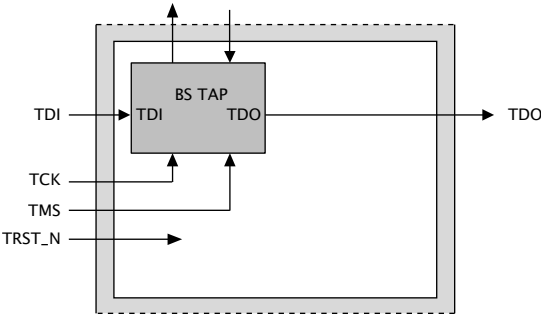


Figure 13:
JTAG chain
structure

The JTAG chain structure is illustrated in Figure 13. It comprises a single 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. It has a 4-bit IR and 32-bit DR. It also provides access to a chip TAP that in turn can access the xCORE Tile for loading code and debugging.

The TRST_N pin must be asserted low during and after power up for 100 ns. If JTAG is not required, the TRST_N pin can be tied to ground to hold the JTAG module in reset.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 14.

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	1	1	0	0	1	1	0	0	0	1	1	0	0	1	1	
0				0				0				0				6				6				3				3				

Figure 14:
IDCODE
return value

13.5 Power Consumption

Figure 23:
xCORE Tile
currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
I(DDCQ)	Quiescent VDD current		45		mA	A, B, C
PD	Tile power dissipation		325		μW/MIPS	A, D, E, F
IDD	Active VDD current		570	700	mA	A, G
I(ADDPLL)	PLL_AVDD current		5	7	mA	H
I(VDD33)	VDD33 current		26.7		mA	I
I(USB_VDD)	USB_VDD current		8.27		mA	J

A Use for budgetary purposes only.

B Assumes typical tile and I/O voltages with no switching activity.

C Includes PLL current.

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

E Assumes 1 MHz = 1 MIPS.

F PD(TYP) value is the usage power consumption under typical operating conditions.

G Measurement conditions: VDD = 1.0 V, VDDIO = 3.3 V, 25 °C, 500 MHz, average device resource usage.

H PLL_AVDD = 1.0 V

I HS mode transmitting while driving all 0's data (constant JKJK on DP/DM). Loading of 10 pF. Transfers do not include any interpacket delay.

J HS receive mode; no traffic.



The tile power consumption of the device is highly application dependent and should be used for budgetary purposes only.

More detailed power analysis can be found in the XS1-UF Power Consumption document,

13.6 Clock

Figure 24:
Clock

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f	Frequency	9	24	25	MHz	
SR	Slew rate	0.10			V/ns	
TJ(LT)	Long term jitter (pk-pk)			2	%	A
f(MAX)	Processor clock frequency			500	MHz	B

A Percentage of CLK period.

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

Further details can be found in the XS1-UF Clock Frequency Control document,

Appendices

A Configuration of the XUF212-512-TQ128

The device is configured through banks of registers, as shown in Figure 30.

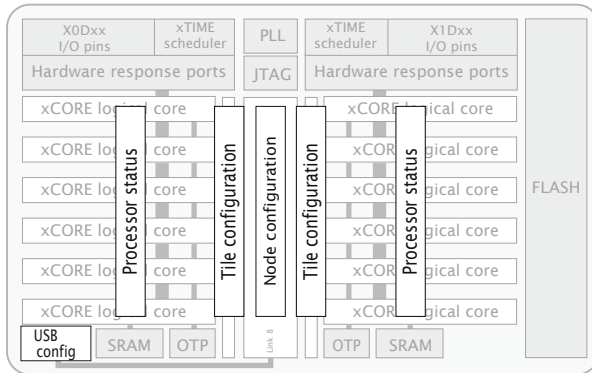


Figure 30:
Registers

The following communication sequences specify how to access those registers. Any messages transmitted contain the most significant 24 bits of the channel-end to which a response is to be sent. This comprises the node-identifier and the channel number within the node. If no response is required on a write operation, supply 24-bits with the last 8-bits set, which suppresses the reply message. Any multi-byte data is sent most significant byte first.

A.1 Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0B. Alternatively, the functions `getps(reg)` and `setps(reg,value)` can be used from XC.

A.2 Accessing an xCORE Tile configuration register

xCORE Tile configuration registers can be accessed through the interconnect using the functions `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tile ↪ ref, ...)`, where `tileref` is the name of the xCORE Tile, e.g. `tile[1]`. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the xCORE tile configuration registers. The destination of the channel-end should be set to `0xnnnnC20C` where `nnnnnn` is the tile-identifier.

A write message comprises the following:

B.1 RAM base address: 0x00

This register contains the base address of the RAM. It is initialized to 0x00040000.

0x00: RAM base address	Bits	Perm	Init	Description
	31:2	RW		Most significant 16 bits of all addresses.
	1:0	RO	-	Reserved

B.2 Vector base address: 0x01

Base address of event vectors in each resource. On an interrupt or event, the 16 most significant bits of the destination address are provided by this register; the least significant 16 bits come from the event vector.

0x01: Vector base address	Bits	Perm	Init	Description
	31:18	RW		The event and interrupt vectors.
	17:0	RO	-	Reserved

B.3 xCORE Tile control: 0x02

Register to control features in the xCORE tile

0x0C: RAM size	Bits	Perm	Init	Description
	31:2	RO		Most significant 16 bits of all addresses.
	1:0	RO	-	Reserved

B.12 Debug SSR: 0x10

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

0x10: Debug SSR	Bits	Perm	Init	Description
	31:11	RO	-	Reserved
	10	DRW		Address space identifier
	9	DRW		Determines the issue mode (DI bit) upon Kernel Entry after Exception or Interrupt.
	8	RO		Determines the issue mode (DI bit).
	7	DRW		When 1 the thread is in fast mode and will continually issue.
	6	DRW		When 1 the thread is paused waiting for events, a lock or another resource.
	5	RO	-	Reserved
	4	DRW		1 when in kernel mode.
	3	DRW		1 when in an interrupt handler.
	2	DRW		1 when in an event enabling sequence.
	1	DRW		When 1 interrupts are enabled for the thread.
	0	DRW		When 1 events are enabled for the thread.

B.13 Debug SPC: 0x11

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

0x11: Debug SPC	Bits	Perm	Init	Description
	31:0	DRW		Value.

B.14 Debug SSP: 0x12

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

0x12: Debug SSP	Bits	Perm	Init	Description
	31:0	DRW		Value.

B.15 DGETREG operand 1: 0x13

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

0x13: DGETREG operand 1	Bits	Perm	Init	Description
	31:8	RO	-	Reserved
	7:0	DRW		Thread number to be read

B.16 DGETREG operand 2: 0x14

Register number to be read by DGETREG

0x14: DGETREG operand 2	Bits	Perm	Init	Description
	31:5	RO	-	Reserved
	4:0	DRW		Register number to be read

B.17 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

0x15: Debug interrupt type	Bits	Perm	Init	Description
	31:18	RO	-	Reserved
	17:16	DRW		Number of the hardware breakpoint/watchpoint which caused the interrupt (always 0 for =HOST= and =DCALL=). If multiple breakpoints/watchpoints trigger at once, the lowest number is taken.
	15:8	DRW		Number of thread which caused the debug interrupt (always 0 in the case of =HOST=).
	7:3	RO	-	Reserved
	2:0	DRW	0	Indicates the cause of the debug interrupt 1: Host initiated a debug interrupt through JTAG 2: Program executed a DCALL instruction 3: Instruction breakpoint 4: Data watch point 5: Resource watch point

0x07:
Security
configuration

Bits	Perm	Init	Description
31	CRO		Disables write permission on this register
30:15	RO	-	Reserved
14	CRO		Disable access to XCore's global debug
13	RO	-	Reserved
12	CRO		lock all OTP sectors
11:8	CRO		lock bit for each OTP sector
7	CRO		Enable OTP redundancy
6	RO	-	Reserved
5	CRO		Override boot mode and read boot image from OTP
4	CRO		Disable JTAG access to the PLL/BOOT configuration registers
3:1	RO	-	Reserved
0	CRO		Disable access to XCore's JTAG debug TAP

C.8 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over the switch. This is the same set of registers as the [Debug Scratch registers in the processor status](#).

0x20 .. 0x27:
Debug
scratch

Bits	Perm	Init	Description
31:0	CRW		Value.

C.9 PC of logical core 0: 0x40

Value of the PC of logical core 0.

0x40:
PC of logical
core 0

Bits	Perm	Init	Description
31:0	CRO		Value.

C.10 PC of logical core 1: 0x41

Value of the PC of logical core 1.

0x0D: Directions 8-15	Bits	Perm	Init	Description
	31:28	RW	0	The direction for packets whose dimension is F.
	27:24	RW	0	The direction for packets whose dimension is E.
	23:20	RW	0	The direction for packets whose dimension is D.
	19:16	RW	0	The direction for packets whose dimension is C.
	15:12	RW	0	The direction for packets whose dimension is B.
	11:8	RW	0	The direction for packets whose dimension is A.
	7:4	RW	0	The direction for packets whose dimension is 9.
	3:0	RW	0	The direction for packets whose dimension is 8.

D.12 Reserved: 0x10

Reserved.

0x10: Reserved	Bits	Perm	Init	Description
	31:2	RO	-	Reserved
	1	RW	0	Reserved.
	0	RW	0	Reserved.

D.13 Reserved.: 0x11

Reserved.

0x11: Reserved.	Bits	Perm	Init	Description
	31:2	RO	-	Reserved
	1	RW	0	Reserved.
	0	RW	0	Reserved.

D.14 Debug source: 0x1F

Contains the source of the most recent debug event.

0x1F:
Debug source

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		Reserved.
3:2	RO	-	Reserved
1	RW		If set, XCore1 is the source of last GlobalDebug event.
0	RW		If set, XCore0 is the source of last GlobalDebug event.

D.15 Link status, direction, and network: 0x20 .. 0x28

These registers contain status information for low level debugging (read-only), the network number that each link belongs to, and the direction that each link is part of. The registers control links 0..7.

0x20 .. 0x28:
Link status,
direction, and
network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		Identify the SRC_TARGET type 0 - SLink, 1 - PLink, 2 - SSCTL, 3 - Undefine.
23:16	RO		When the link is in use, this is the destination link number to which all packets are sent.
15:12	RO	-	Reserved
11:8	RW	0	The direction that this link operates in.
7:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, reset as 0.
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO		1 when the dest side of the link is in use.
0	RO		1 when the source side of the link is in use.

D.16 PLink status and network: 0x40 .. 0x47

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

0x40 .. 0x47:
PLink status
and network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		Identify the SRC_TARGET type 0 - SLink, 1 - PLink, 2 - SSCTL, 3 - Undefine.
23:16	RO		When the link is in use, this is the destination link number to which all packets are sent.
15:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, reset as 0.
3	RO	-	Reserved
2	RO		1 when the current packet is considered junk and will be thrown away.
1	RO		1 when the dest side of the link is in use.
0	RO		1 when the source side of the link is in use.

D.17 Link configuration and initialization: 0x80 .. 0x88

These registers contain configuration and debugging information specific to external links. The link speed and width can be set, the link can be initialized, and the link status can be monitored. The registers control links 0..7.

0x80 .. 0x88:
Link
configuration
and
initialization

Bits	Perm	Init	Description
31	RW		Write to this bit with '1' will enable the XLink, writing '0' will disable it. This bit controls the muxing of ports with overlapping xlinks.
30	RW	0	0: operate in 2 wire mode; 1: operate in 5 wire mode
29:28	RO	-	Reserved
27	RO		Rx buffer overflow or illegal token encoding received.
26	RO	0	This end of the xlink has issued credit to allow the remote end to transmit
25	RO	0	This end of the xlink has credit to allow it to transmit.
24	WO		Clear this end of the xlink's credit and issue a HELLO token.
23	WO		Reset the receiver. The next symbol that is detected will be the first symbol in a token.
22	RO	-	Reserved
21:11	RW	0	Specify min. number of idle system clocks between two continuous symbols within a transmit token -1.
10:0	RW	0	Specify min. number of idle system clocks between two continuous transmit tokens -1.

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

G.3 Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section G.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 XL0, XL1, etc in the function column. The 2-wire link comprises two inputs and outputs, labelled $XL0_{out}^1$, $XL0_{out}^0$, $XL0_{in}^0$, and $XL0_{in}^1$. For example, if you choose to use XL0 for xSCOPE I/O, you need to connect up $XL0_{out}^1$, $XL0_{out}^0$, $XL0_{in}^0$, $XL0_{in}^1$ as follows:

- ▶ $XL0_{out}^1$ (X0D43) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- ▶ $XL0_{out}^0$ (X0D42) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- ▶ $XL0_{in}^0$ (X0D41) to pin 14 of the xSYS header.
- ▶ $XL0_{in}^1$ (X0D40) to pin 18 of the xSYS header.