



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

#### Details

Product Status	Active
Core Processor	XCore
Core Size	32-Bit 16-Core
Speed	2000MIPS
Connectivity	-
Peripherals	-
Number of I/O	88
Program Memory Size	-
Program Memory Type	ROMIess
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/xl216-512-tq128-i20

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# 1 xCORE Multicore Microcontrollers

The xCORE-200 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: XL216-512-TQ128 block diagram

Key features of the XL216-512-TQ128 include:

- ► **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between five 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 6.1
- ▶ **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 6.2
- 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 6.5
- 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 6.6

-XM()S

- ▶ **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 6.3
- Clock blocks xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 6.4
- 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 9
- PLL The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 7
- JTAG The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 10

#### 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 XL216-512-TQ128 Features

#### ► Multicore Microcontroller with Advanced Multi-Core RISC Architecture

- 16 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 1/5 and 1/8 of tile MIPS
  - 16x32bit dedicated registers
- 167 high-density 16/32-bit instructions
  - All have single clock-cycle execution (except for divide)
  - 32x32 $\rightarrow$ 64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions

#### Programmable I/O

- 88 general-purpose I/O pins, configurable as input or output
  - Up to 32 x 1bit port, 12 x 4bit port, 8 x 8bit port, 4 x 16bit port
  - 4 xCONNECT links
- Port sampling rates of up to 60 MHz with respect to an external clock
- 64 channel endss (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

#### 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

-XM()S

- 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



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. xCORE-200 clock blocks optionally divide the clock input from a 1-bit port.

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 xCORE Tile reference clock.

## 6.5 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.

## 6.6 xCONNECT Switch and Links

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

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

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

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. When the device is taken out of reset by releasing RST\_N the processor starts its internal reset process. After 15-150  $\mu$ s (depending on the input clock) the processor boots.

The xCORE Tile boot procedure is illustrated in Figure 8. If bit 5 of the security register (*see* §9.1) is set, the device boots from OTP. To get a high value, a 3K3 pull-up resistor should be strapped onto the pin. To assure a low value, a pull-down resistor is required if other external devices are connected to this port.



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.

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.

#### 8.6 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 13 provide a strong level of protection and are sufficient for providing strong IP security.

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 ( <i>see</i> §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.
	2115	General purpose software accessable security register available to end-users.
	3122	General purpose user programmable JTAG UserID code extension.

-XM()S

Figure 13: Security register features 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.

Symbol	Parameter	MIN	ТҮР	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			18	MHz	
f(TCK_B)	TCK frequency (boundary scan)			10	MHz	
T(SETUP)	TDO to TCK setup time	5			ns	А
T(HOLD)	TDO to TCK hold time	5			ns	А
T(DELAY)	TCK to output delay			15	ns	В

## 12.9 JTAG Timing

Figure 26: JTAG timing

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 apart from the global asynchronous reset TRST\_N.



# Appendices

# A Configuration of the XL216-512-TQ128

The device is configured through banks of registers, as shown in Figure 29.



Figure 29: 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  $\rightarrow$  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** Processor Status Configuration

The processor status control registers can be accessed directly by the processor using processor status reads and writes (use getps(reg) and setps(reg,value) for reads and writes).

Number	Perm	Description
0x00	RW	RAM base address
0x01	RW	Vector base address
0x02	RW	xCORE Tile control
0x03	RO	xCORE Tile boot status
0x05	RW	Security configuration
0x06	RW	Ring Oscillator Control
0x07	RO	Ring Oscillator Value
0x08	RO	Ring Oscillator Value
0x09	RO	Ring Oscillator Value
0x0A	RO	Ring Oscillator Value
0x0C	RO	RAM size
0x10	DRW	Debug SSR
0x11	DRW	Debug SPC
0x12	DRW	Debug SSP
0x13	DRW	DGETREG operand 1
0x14	DRW	DGETREG operand 2
0x15	DRW	Debug interrupt type
0x16	DRW	Debug interrupt data
0x18	DRW	Debug core control
0x20 0x27	DRW	Debug scratch
0x30 0x33	DRW	Instruction breakpoint address
0x40 0x43	DRW	Instruction breakpoint control
0x50 0x53	DRW	Data watchpoint address 1
0x60 0x63	DRW	Data watchpoint address 2
0x70 0x73	DRW	Data breakpoint control register
0x80 0x83	DRW	Resources breakpoint mask
0x90 0x93	DRW	Resources breakpoint value
0x9C 0x9F	DRW	Resources breakpoint control register

-XMOS°

Figure 30: Summary

XS2-L16A-512-TQ128

#### B.4 xCORE Tile boot status: 0x03

This read-only register describes the boot status of the xCORE tile.

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Processor number.
15:9	RO	-	Reserved
8	RO		Overwrite BOOT_MODE.
7:6	RO	-	Reserved
5	RO		Indicates if core1 has been powered off
4	RO		Cause the ROM to not poll the OTP for correct read levels
3	RO		Boot ROM boots from RAM
2	RO		Boot ROM boots from JTAG
1:0	RO		The boot PLL mode pin value.

0x03: xCORE Tile boot status

## B.5 Security configuration: 0x05

Copy of the security register as read from OTP.

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

**0x05:** Security configuration

0x13	Bits	Perm	Init	Description
DGETREG	31:8	RO	-	Reserved
operand 1	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
2	4:0	DRW		Register number to be read

#### B.17 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

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

**0x15:** Debug interrupt type

## B.18 Debug interrupt data: 0x16

On a data watchpoint, this register contains the effective address of the memory operation that triggered the debugger. On a resource watchpoint, it countains the resource identifier.

-XMOS-

Ox16:<br/>Debug<br/>interrupt dataBitsPermInitDescription31:0DRWValue.

#### B.19 Debug core control: 0x18

This register enables the debugger to temporarily disable logical cores. When returning from the debug interrupts, the cores set in this register will not execute. This enables single stepping to be implemented.

0x18: Debug core control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		1-hot vector defining which threads are stopped when not in debug mode. Every bit which is set prevents the respective thread from running.

#### B.20 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

ug	Bits	Perm	Init	Description
tch	31:0	DRW		Value.

#### B.21 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

tion bint	Bits	Perm	Init	Description
ess	31:0	DRW		Value.

-XM()S

### B.22 Instruction breakpoint control: 0x40 .. 0x43

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

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:2	RO	-	Reserved
1	DRW	0	When 0 break when PC == IBREAK_ADDR. When 1 = break when PC != IBREAK_ADDR.
0	DRW	0	When 1 the instruction breakpoint is enabled.

0x40 .. 0x43: Instruction breakpoint control

### B.23 Data watchpoint address 1: 0x50 .. 0x53

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

**0x50 .. 0x53:** Data watchpoint address 1

Data point	Bits	Perm	Init	Description
ess 1	31:0	DRW		Value.

#### B.24 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

Data 1point	Bits	Perm	Init	Description
ress 2	31:0	DRW		Value.

## B.25 Data breakpoint control register: 0x70 .. 0x73

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

-XMOS

XS2-L16A-512-TQ128

	Bits	Perm	Init	Description
	31:24	RO	-	Reserved
	23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
0x70 0x73:	15:3	RO	-	Reserved
Data break point	2	DRW	0	When 1 the breakpoints will be be triggered on loads.
control	1	DRW	0	Determines the break condition: $0 = A AND B$ , $1 = A OR B$ .
register	0	DRW	0	When 1 the instruction breakpoint is enabled.

#### B.26 Resources breakpoint mask: 0x80 .. 0x83

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

0x80 0x83:				
breakpoint	Bits	Perm	Init	Description
mask	31:0	DRW		Value.

#### B.27 Resources breakpoint value: 0x90 .. 0x93

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

0x90 0x93:				
breakpoint	Bits	Perm	Init	Description
value	31:0	DRW		Value.

#### B.28 Resources breakpoint control register: 0x9C .. 0x9F

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

-XMOS

40

# C Tile Configuration

The xCORE Tile control registers can be accessed using configuration reads and writes (use write\_tile\_config\_reg(tileref, ...) and read\_tile\_config\_reg(tileref,  $\rightarrow$  ...) for reads and writes).

Number	Perm	Description
0x00	CRO	Device identification
0x01	CRO	xCORE Tile description 1
0x02	CRO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	CRW	xCORE Tile clock divider
0x07	CRO	Security configuration
0x20 0x27	CRW	Debug scratch
0x40	CRO	PC of logical core 0
0x41	CRO	PC of logical core 1
0x42	CRO	PC of logical core 2
0x43	CRO	PC of logical core 3
0x44	CRO	PC of logical core 4
0x45	CRO	PC of logical core 5
0x46	CRO	PC of logical core 6
0x47	CRO	PC of logical core 7
0x60	CRO	SR of logical core 0
0x61	CRO	SR of logical core 1
0x62	CRO	SR of logical core 2
0x63	CRO	SR of logical core 3
0x64	CRO	SR of logical core 4
0x65	CRO	SR of logical core 5
0x66	CRO	SR of logical core 6
0x67	CRO	SR of logical core 7

Figure 31: Summary

## C.1 Device identification: 0x00

This register identifies the xCORE Tile

-XMOS°

0x04: Control PSwitch permissions to debug registers

Bits	Perm	Init	Description
31	CRW	0	When 1 the PSwitch is restricted to RO access to all CRW registers from SSwitch, XCore(PS_DBG_Scratch) and JTAG
30:1	RO	-	Reserved
0	CRW	0	When 1 the PSwitch is restricted to RO access to all CRW registers from SSwitch

## C.5 Cause debug interrupts: 0x05

This register can be used to raise a debug interrupt in this xCORE tile.

**0x05:** Cause debug interrupts

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	CRW	0	1 when the processor is in debug mode.
0	CRW	0	Request a debug interrupt on the processor.

## C.6 xCORE Tile clock divider: 0x06

This register contains the value used to divide the PLL clock to create the xCORE tile clock. The divider is enabled under control of the tile control register

0x06: xCORE Tile clock divider

Bits	Perm	Init	Description
31	CRW	0	Clock disable. Writing '1' will remove the clock to the tile.
30:16	RO	-	Reserved
15:0	CRW	0	Clock divider.

## C.7 Security configuration: 0x07

Copy of the security register as read from OTP.

	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 reduanacy
	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
1	0	CRO		Disable access to XCore's JTAG debug TAP

0x07 Security configuration

### 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

ebug	Bits	Perm	Init	Description
ratch	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

ical	Bits	Perm	Init	Description
re O	31:0	CRO		Value.

-XMOS

## C.10 PC of logical core 1: 0x41

Value of the PC of logical core 1.



**0x62:** SR of logical core 2

Bits	Perm	Init	Description
31:0	CRO		Value.

## C.20 SR of logical core 3: 0x63

Value of the SR of logical core 3

0x63: SR of logical core 3

Bits	Perm	Init	Description
31:0	CRO		Value.

## C.21 SR of logical core 4: 0x64

Value of the SR of logical core 4

Ox64:<br/>SR of logical<br/>core 4BitsPermInitDescription31:0CROValue.

### C.22 SR of logical core 5: 0x65

Value of the SR of logical core 5

**0x65** SR of logical core 5

<b>x65:</b> gical	Bits	Perm	Init	Description
re 5	31:0	CRO		Value.

## C.23 SR of logical core 6: 0x66

Value of the SR of logical core 6

**0x66:** SR of logical core 6

Bits	Perm	Init	Description
31:0	CRO		Value.

# F Schematics Design Check List

✓ This section is a checklist for use by schematics designers using the XL216-512-TQ128. Each of the following sections contains items to check for each design.

### F.1 Power supplies

- □ VDDIO and OTP\_VCC supply is within specification before the VDD (core) supply is turned on. Specifically, the VDDIO and OTP\_VCC supply is within specification before VDD (core) reaches 0.4V (Section 11).
- The VDD (core) supply ramps monotonically (rises constantly) from 0V to its final value (0.95V 1.05V) within 10ms (Section 11).
- The VDD (core) supply is capable of supplying 700 mA (Section 11 and Figure 18).
- PLL\_AVDD is filtered with a low pass filter, for example an RC filter, see Section 11

#### F.2 Power supply decoupling

- The design has multiple decoupling capacitors per supply, for example at least four0402 or 0603 size surface mount capacitors of 100nF in value, per supply (Section 11).
- A bulk decoupling capacitor of at least 10uF is placed on each supply (Section 11).

#### F.3 Power on reset

□ The RST\_N and TRST\_N pins are asserted (low) during or after power up. The device is not used until these resets have taken place.

#### F.4 Clock

- The CLK input pin is supplied with a clock with monotonic rising edges and low jitter.
- You have chosen an input clock frequency that is supported by the device (Section 7).

# H Associated Design Documentation

Document Title	Information	Document Number
Estimating Power Consumption For XS1-L Devices	Power consumption	X4271
Programming XC on XMOS Devices	Timers, ports, clocks, cores and channels	X9577
xTIMEcomposer User Guide	Compilers, assembler and linker/mapper	X3766
	Timing analyzer, xScope, debugger	
	Flash and OTP programming utilities	

# I Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	X7879
XS1 Port I/O Timing	Port timings	X5821
xCONNECT Architecture	Link, switch and system information	X4249
XS1-L Link Performance and Design Guidelines	Link timings	X2999
XS1-L Clock Frequency Control	Advanced clock control	X1433
XS1-L Active Power Conservation	Low-power mode during idle	X7411

-XMOS<sup>®</sup>-

63