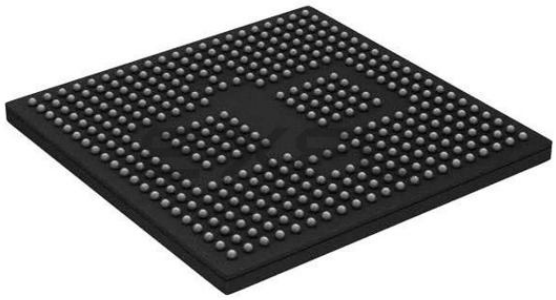


Welcome to E-XFL.COM

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 24-Core
Speed	4000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	176
Program Memory Size	-
Program Memory Type	ROMless
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	374-LFBGA
Supplier Device Package	374-FBGA (18x18)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xu224-512-fb374-i40

4 Signal Description

This section lists the signals and I/O pins available on the XU224-512-FB374. 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.
- IOT: The IO pin is powered from VDDIOT (X1) or VDDIOT_2 (X3), not VDDIO
- IO: the pin is powered from VDDIO

Power pins (12)			
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_2_VDD	Digital tile power	PWR	
USB_2_VDD33	USB Analog power	PWR	
USB_VDD	Digital tile power	PWR	
USB_VDD33	USB Analog power	PWR	
VDD	Digital tile power	PWR	
VDDIO	Digital I/O power	PWR	
VDDIOT	Digital I/O power (top)	PWR	
VDDIOT_2	Digital I/O power (top, X3)	PWR	

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

(continued)

Signal	Function	Type	Properties
TRST_N	Test reset input	Input	IO, PU, ST

I/O pins (176)								
Signal	**Function**						**Type**	**Properties**
X0D00		$1A^0$					I/O	IO, PD
X0D01	$X_0L3^2_{out}$	$1B^0$					I/O	IO, PD
X0D02			$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	IO, PD
X0D03			$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	I/O	IO, PD
X0D04			$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O	IO, PD
X0D05			$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O	IO, PD
X0D06			$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O	IO, PD
X0D07			$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O	IO, PD
X0D08			$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	IO, PD
X0D09			$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	IO, PD
X0D10	$X_0L3^3_{out}$	$1C^0$					I/O	IO, PD
X0D11		$1D^0$					I/O	IO, PD
X0D12		$1E^0$					I/O	IO, PD
X0D13		$1F^0$					I/O	IO, PD
X0D14			$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	IO, PD
X0D15			$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	IO, PD
X0D20			$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	IO, PD
X0D21			$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	IO, PD
X0D22		$1G^0$					I/O	IO, PD
X0D23		$1H^0$					I/O	IO, PD
X0D28			$4F^0$	$8C^2$	$16B^2$		I/O	IO, PD
X0D29			$4F^1$	$8C^3$	$16B^3$		I/O	IO, PD
X0D30			$4F^2$	$8C^4$	$16B^4$		I/O	IO, PD
X0D31			$4F^3$	$8C^5$	$16B^5$		I/O	IO, PD
X0D32			$4E^2$	$8C^6$	$16B^6$		I/O	IO, PD
X0D33			$4E^3$	$8C^7$	$16B^7$		I/O	IO, PD
X0D36		$1M^0$		$8D^0$	$16B^8$		I/O	IO, PD
X0D37	$X_0L0^4_{in}$	$1N^0$		$8D^1$	$16B^9$		I/O	IO, PD
X0D38	$X_0L0^3_{in}$	$1O^0$		$8D^2$	$16B^{10}$		I/O	IO, PD
X0D39	$X_0L0^2_{in}$	$1P^0$		$8D^3$	$16B^{11}$		I/O	IO, PD
X0D40	$X_0L0^1_{in}$			$8D^4$	$16B^{12}$		I/O	IO, PD
X0D41	$X_0L0^0_{in}$			$8D^5$	$16B^{13}$		I/O	IO, PD
X0D42	$X_0L0^0_{out}$			$8D^6$	$16B^{14}$		I/O	IO, PD
X0D43	$X_0L0^1_{out}$			$8D^7$	$16B^{15}$		I/O	IO, PD
X1D10		$1C^0$					I/O	IOT, PD
X1D11		$1D^0$					I/O	IOT, PD
X1D12		$1E^0$					I/O	IO, PD
X1D13		$1F^0$					I/O	IO, PD

(continued)

Signal	Function						Type	Properties
X1D14			4C[0]	8B[0]	16A[8]	32A[28]	I/O	IO, PD
X1D15			4C[1]	8B[1]	16A[9]	32A[29]	I/O	IO, PD
X1D16	$X_0L3^1_{in}$		4D[0]	8B[2]	16A[10]		I/O	IO, PD
X1D17	$X_0L3^0_{in}$		4D[1]	8B[3]	16A[11]		I/O	IO, PD
X1D18	$X_0L3^0_{out}$		4D[2]	8B[4]	16A[12]		I/O	IO, PD
X1D19	$X_0L3^1_{out}$		4D[3]	8B[5]	16A[13]		I/O	IO, PD
X1D20			4C[2]	8B[6]	16A[14]	32A[30]	I/O	IO, PD
X1D21			4C[3]	8B[7]	16A[15]	32A[31]	I/O	IO, PD
X1D22	$X_0L3^4_{out}$	1G[0]					I/O	IO, PD
X1D23		1H[0]					I/O	IO, PD
X1D24		1I[0]					I/O	IO, PD
X1D25		1J[0]					I/O	IO, 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
X1D34	$X_0L0^2_{out}$	1K[0]					I/O	IO, PD
X1D35	$X_0L0^3_{out}$	1L[0]					I/O	IO, PD
X1D36	$X_0L0^4_{out}$	1M[0]		8D[0]	16B[8]		I/O	IO, PD
X1D37	$X_0L3^4_{in}$	1N[0]		8D[1]	16B[9]		I/O	IO, PD
X1D38	$X_0L3^3_{in}$	1O[0]		8D[2]	16B[10]		I/O	IO, PD
X1D39	$X_0L3^2_{in}$	1P[0]		8D[3]	16B[11]		I/O	IO, 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
X1D49	$X_0L1^4_{in}$					32A[0]	I/O	IO, PD
X1D50	$X_0L1^3_{in}$					32A[1]	I/O	IO, PD
X1D51	$X_0L1^2_{in}$					32A[2]	I/O	IO, PD
X1D52	$X_0L1^1_{in}$					32A[3]	I/O	IO, PD
X1D53	$X_0L1^0_{in}$					32A[4]	I/O	IO, PD
X1D54	$X_0L1^0_{out}$					32A[5]	I/O	IO, PD
X1D55	$X_0L1^1_{out}$					32A[6]	I/O	IO, PD
X1D56	$X_0L1^2_{out}$					32A[7]	I/O	IO, PD
X1D57	$X_0L1^3_{out}$					32A[8]	I/O	IO, PD
X1D58	$X_0L1^4_{out}$					32A[9]	I/O	IO, PD
X1D61	$X_0L2^4_{in}$					32A[10]	I/O	IO, PD
X1D62	$X_0L2^3_{in}$					32A[11]	I/O	IO, PD
X1D63	$X_0L2^2_{in}$					32A[12]	I/O	IO, PD

(continued)

Signal	Function	Type	Properties
X2D50	$X_2L5^3_{in}$ $32A^1$	I/O	IO, PD
X2D51	$X_2L5^2_{in}$ $32A^2$	I/O	IO, PD
X2D52	$X_2L5^1_{in}$ $32A^3$	I/O	IO, PD
X2D53	$X_2L5^0_{in}$ $32A^4$	I/O	IO, PD
X2D54	$X_2L5^0_{out}$ $32A^5$	I/O	IO, PD
X2D55	$X_2L5^1_{out}$ $32A^6$	I/O	IO, PD
X2D56	$X_2L5^2_{out}$ $32A^7$	I/O	IO, PD
X2D57	$X_2L5^3_{out}$ $32A^8$	I/O	IO, PD
X2D58	$X_2L5^4_{out}$ $32A^9$	I/O	IO, PD
X2D61	$X_2L6^4_{in}$ $32A^{10}$	I/O	IO, PD
X2D62	$X_2L6^3_{in}$ $32A^{11}$	I/O	IO, PD
X2D63	$X_2L6^2_{in}$ $32A^{12}$	I/O	IO, PD
X2D64	$X_2L6^1_{in}$ $32A^{13}$	I/O	IO, PD
X2D65	$X_2L6^0_{in}$ $32A^{14}$	I/O	IO, PD
X2D66	$X_2L6^0_{out}$ $32A^{15}$	I/O	IO, PD
X2D67	$X_2L6^1_{out}$ $32A^{16}$	I/O	IO, PD
X2D68	$X_2L6^2_{out}$ $32A^{17}$	I/O	IO, PD
X2D69	$X_2L6^3_{out}$ $32A^{18}$	I/O	IO, PD
X2D70	$X_2L6^4_{out}$ $32A^{19}$	I/O	IO, PD
X3D00	$X_2L7^2_{in}$ $1A^0$	I/O	IO, PD
X3D01	$X_2L7^1_{in}$ $1B^0$	I/O	IO, PD
X3D02	$X_2L4^0_{in}$ $4A^0$ $8A^0$ $16A^0$ $32A^{20}$	I/O	IO, PD
X3D03	$X_2L4^0_{out}$ $4A^1$ $8A^1$ $16A^1$ $32A^{21}$	I/O	IO, PD
X3D04	$X_2L4^1_{out}$ $4B^0$ $8A^2$ $16A^2$ $32A^{22}$	I/O	IO, PD
X3D05	$X_2L4^2_{out}$ $4B^1$ $8A^3$ $16A^3$ $32A^{23}$	I/O	IO, PD
X3D06	$X_2L4^3_{out}$ $4B^2$ $8A^4$ $16A^4$ $32A^{24}$	I/O	IO, PD
X3D07	$X_2L4^4_{out}$ $4B^3$ $8A^5$ $16A^5$ $32A^{25}$	I/O	IO, PD
X3D08	$X_2L7^4_{in}$ $4A^2$ $8A^6$ $16A^6$ $32A^{26}$	I/O	IO, PD
X3D09	$X_2L7^3_{in}$ $4A^3$ $8A^7$ $16A^7$ $32A^{27}$	I/O	IO, PD
X3D10	$1C^0$	I/O	IOT, PD
X3D11	$1D^0$	I/O	IOT, PD
X3D12	$1E^0$	I/O	IO, PD
X3D13	$1F^0$	I/O	IO, PD
X3D14	$4C^0$ $8B^0$ $16A^8$ $32A^{28}$	I/O	IO, PD
X3D15	$4C^1$ $8B^1$ $16A^9$ $32A^{29}$	I/O	IO, PD
X3D20	$4C^2$ $8B^6$ $16A^{14}$ $32A^{30}$	I/O	IO, PD
X3D21	$4C^3$ $8B^7$ $16A^{15}$ $32A^{31}$	I/O	IO, PD
X3D23	$1H^0$	I/O	IO, PD
X3D24	$1I^0$	I/O	IO, PD
X3D25	$1J^0$	I/O	IO, PD
X3D26	$4E^0$ $8C^0$ $16B^0$	I/O	IOT, PD
X3D27	$4E^1$ $8C^1$ $16B^1$	I/O	IOT, PD
X3D28	$4F^0$ $8C^2$ $16B^2$	I/O	IOT, PD

(continued)

5 Example Application Diagram

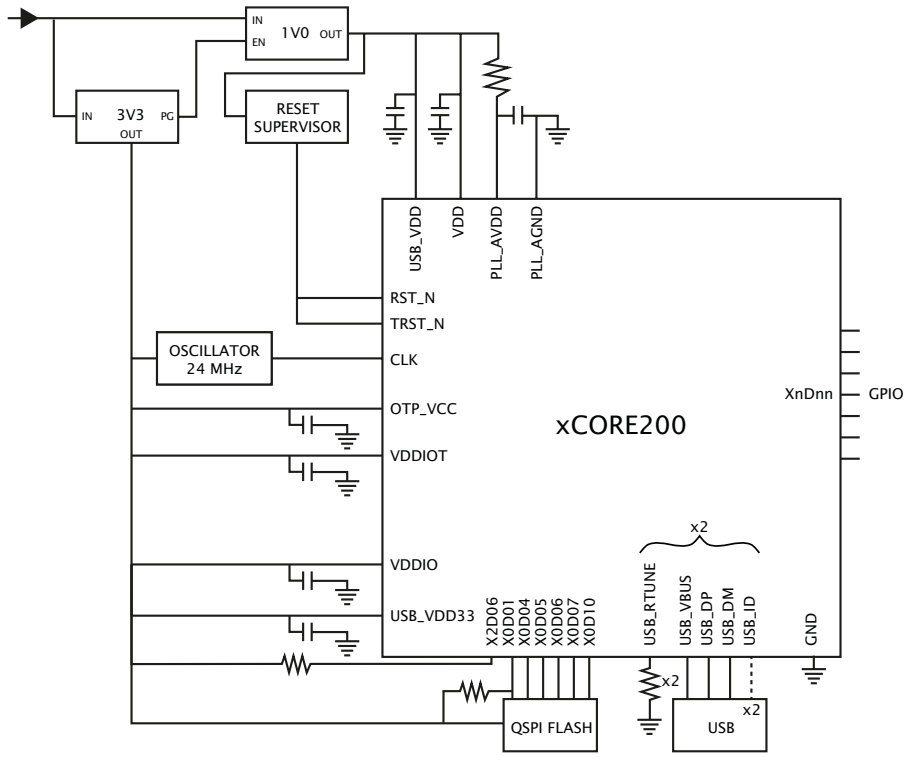


Figure 2: Simplified Reference Schematic

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

ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.

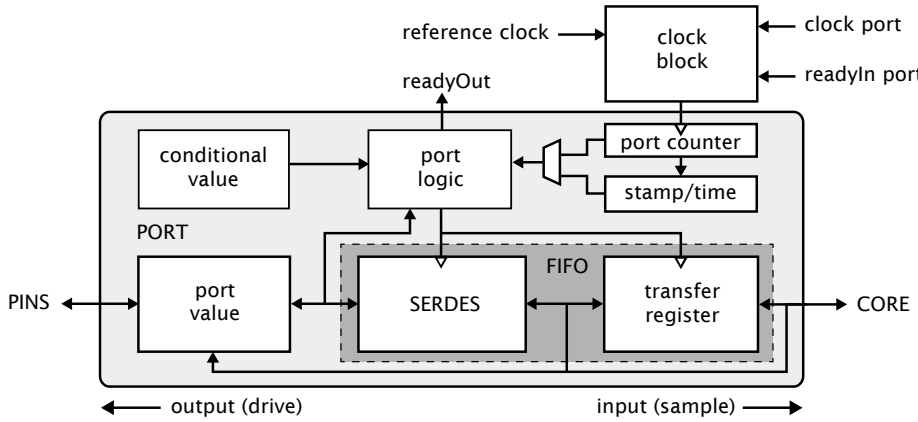


Figure 4: Port block diagram

The port logic can drive its pins high or low, or it can sample the value on its pins, optionally waiting for a particular condition. Ports are accessed using dedicated instructions that are executed in a single processor cycle. xCORE-200 IO pins can be used as *open collector* outputs, where signals are driven low if a zero is output, but left high impedance if a one is output. This option is set on a per-port basis.

Data is transferred between the pins and core using a FIFO that comprises a SERDES and transfer register, providing options for serialization and buffered data.

Each port has a 16-bit counter that can be used to control the time at which data is transferred between the port value and transfer register. The counter values can be obtained at any time to find out when data was obtained, or used to delay I/O until some time in the future. The port counter value is automatically saved as a timestamp, that can be used to provide precise control of response times.

The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

6.4 Clock blocks

xCORE devices include a set of programmable clocks called clock blocks that can be used to govern the rate at which ports execute. Each xCORE tile has six clock blocks: the first clock block provides the tile reference clock and runs at a default frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.

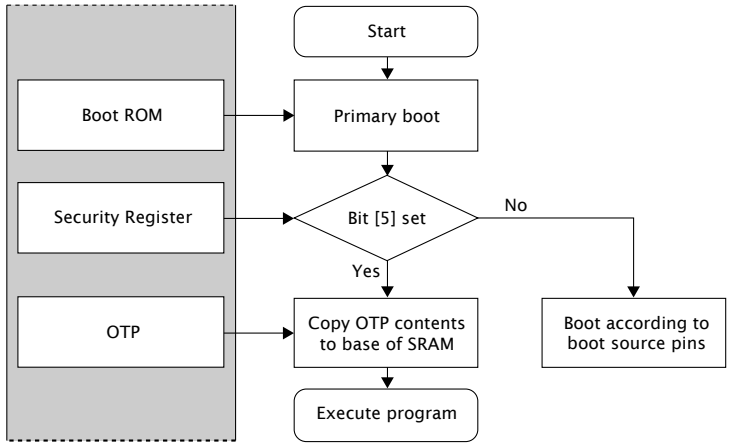


Figure 8: Boot procedure

X0D06	X0D05	X0D04	Tile 0 boot	Tile 1 boot	Enabled links
0	0	0	QSPI master	Channel end 0	None
0	0	1	SPI master	Channel end 0	None
0	1	0	SPI slave	Channel end 0	None
0	1	1	SPI slave	SPI slave	None
1	0	0	Channel end 0	Channel end 0	XL0 (2w)
1	0	1	Channel end 0	Channel end 0	XL4-XL7 (5w)
1	1	0	Channel end 0	Channel end 0	XL1, XL2, XL5, and XL6 (5w)
1	1	1	Channel end 0	Channel end 0	XL0-XL3 (5w)

Figure 9: Boot source pins

8.1 Boot from QSPI master

If set to boot from QSPI master, the processor enables the six pins specified in Figure 10, and drives the SPI clock at 50 MHz (assuming a 400 MHz core clock). A READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

Pin	Signal	Description
X0D01	SS	Slave Select
X0D04..X0D07	SPIO	Data
X0D10	SCLK	Clock

Figure 10: QSPI pins

The xCORE Tile expects each byte to be transferred with the *least-significant nibble first*. Programmers who write bytes into an QSPI interface using the most significant nibble first may have to reverse the nibbles in each byte of the image stored in the QSPI device.

resistors on the link, drives all the TX wires low (the initial state for the Link), and monitors the RX pins for boot-traffic; they must be low at this stage. If the internal pull-down is too weak to drain any residual charge, external pull-downs of 10K may be required on those pins.

The boot-rom on the core will then:

1. Allocate channel-end 0.
2. Input a word on channel-end 0. It will use this word as a channel to acknowledge the boot. Provide the null-channel-end 0x0000FF02 if no acknowledgment is required.
3. Input the boot image specified above, including the CRC.
4. Input an END control token.
5. Output an END control token to the channel-end received in step 2.
6. Free channel-end 0.
7. Jump to the loaded code.

8.5 Boot from OTP

If an xCORE tile is set to use secure boot (see Figure 8), 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.

8.6 Security register

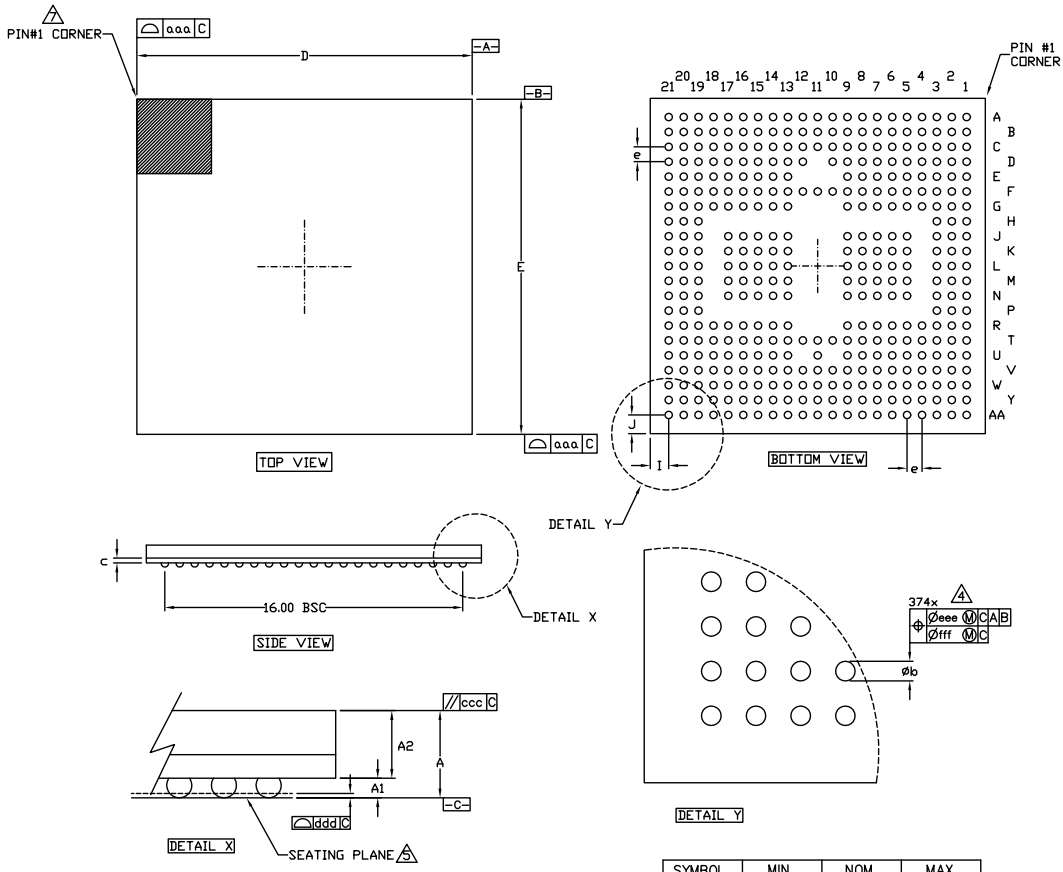
The security register enables security features on the xCORE tile. The features shown in Figure 13 provide a strong level of protection and are sufficient for providing strong IP security.

9 Memory

9.1 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

14 Package Information



SYMBOL	MIN.	NOM.	MAX.
A	1.16	1.26	1.36
A1	0.25	0.30	0.35
A2	0.91	0.96	1.01
D	17.90	18.00	18.10
E	17.90	18.00	18.10
I	1.00 REF.		
J	1.00 REF.		
M	21x21<DEPOPULATED>		
aaa			0.15
ccc			0.20
ddd			0.10
eee			0.15
fff			0.08
b	0.35	0.40	0.45
e	0.80 BSC.		
c	0.26 REF.		

NOTE:

1. ALL DIMENSIONS ARE IN MILLIMETERS.
2. "e" REPRESENTS THE BASIC SOLDER BALL GRID PITCH.
3. "M" REPRESENTS THE MAXIMUM SOLDER BALL MATRIX SIZE.
4. DEMENSIONS "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER PARALLEL TO PRIMARY DATUM C .
5. PRIMARY DATUM C AND SEATING PLANE ARE DESIGNED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
6. DIMENSIONING AND TOLERANCING PER ASME Y14.5M-1994
7. A1 CORNER MUST BE IDENTIFIED BY LASER MARK.

Appendices

A Configuration of the XU224-512-FB374

The device is configured through banks of registers, as shown in Figure 32.

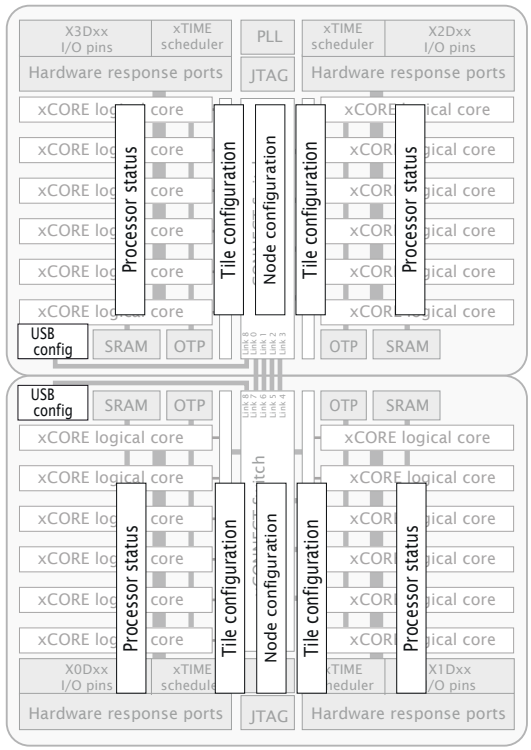


Figure 32: Registers

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.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.

A channel-end should be allocated to communicate with the configuration registers. The destination of the channel-end should be set to `0xnnnnpp02` where `nnnn` is the node-identifier and `pp` is the peripheral identifier.

A write message comprises the following:

control-token 36	24-bit response channel-end identifier	8-bit register number	8-bit size	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 37	24-bit response channel-end identifier	8-bit register number	8-bit size	control-token 1

The response to the read message comprises either control token 3, data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

0x07: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.8 Ring Oscillator Value: 0x08

This register contains the current count of the xCORE Tile Wire ring oscillator. This value is not reset on a system reset.

0x08: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.9 Ring Oscillator Value: 0x09

This register contains the current count of the Peripheral Cell ring oscillator. This value is not reset on a system reset.

0x09: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.10 Ring Oscillator Value: 0x0A

This register contains the current count of the Peripheral Wire ring oscillator. This value is not reset on a system reset.

0x0A: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.11 RAM size: 0x0C

The size of the RAM in bytes

B.25 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 thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:3	RO	-	Reserved
2	DRW	0	When 1 the breakpoints will be be triggered on loads.
1	DRW	0	Determines the break condition: 0 = A AND B, 1 = A OR B.
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: Resources breakpoint mask

Bits	Perm	Init	Description
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: Resources breakpoint value

Bits	Perm	Init	Description
31:0	DRW		Value.

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

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

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

0x64: SR of logical core 4

Bits	Perm	Init	Description
31:0	CRO		Value.

C.22 SR of logical core 5: 0x65

Value of the SR of logical core 5

0x65: SR of logical core 5

Bits	Perm	Init	Description
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.

0x06: PLL settings

Bits	Perm	Init	Description
31	RW		If set to 1, the chip will not be reset
30	RW		If set to 1, the chip will not wait for the PLL to re-lock. Only use this if a gradual change is made to the PLL
29	DW		If set to 1, set the PLL to be bypassed
28	DW		If set to 1, set the boot mode to boot from JTAG
27:26	RO	-	Reserved
25:23	RW		Output divider value range from 1 (8'h0) to 250 (8'hF9). P value.
22:21	RO	-	Reserved
20:8	RW		Feedback multiplication ratio, range from 1 (8'h0) to 255 (8'hFE). M value.
7	RO	-	Reserved
6:0	RW		Oscilator input divider value range from 1 (8'h0) to 32 (8'h0F). N value.

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	SSwitch clock generation

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	Software ref. clock divider

D.18 Static link configuration: 0xA0 .. 0xA7

These registers are used for static (ie, non-routed) links. When a link is made static, all traffic is forwarded to the designated channel end and no routing is attempted. The registers control links C, D, A, B, G, H, E, and F in that order.

0xA0 .. 0xA7: Static link configuration

Bits	Perm	Init	Description
31	RW	0	Enable static forwarding.
30:9	RO	-	Reserved
8	RW	0	The destination processor on this node that packets received in static mode are forwarded to.
7:5	RO	-	Reserved
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.

F.7 UIFM Serial Control: 0x18

0x18: UIFM Serial Control

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RO	0	1 if UIFM is in UTMI+ RXRCV mode.
5	RO	0	1 if UIFM is in UTMI+ RXDM mode.
4	RO	0	1 if UIFM is in UTMI+ RXDP mode.
3	RW	0	Set to 1 to switch UIFM to UTMI+ TXSE0 mode.
2	RW	0	Set to 1 to switch UIFM to UTMI+ TXDATA mode.
1	RW	1	Set to 0 to switch UIFM to UTMI+ TXENABLE mode.
0	RW	0	Set to 1 to switch UIFM to UTMI+ FSLSSERIAL mode.

F.8 UIFM signal flags: 0x1C

Set of flags that monitor line and error states. These flags normally clear on the next packet, but they may be made sticky by using PER_UIFM_FLAGS_STICKY, in which they must be cleared explicitly.

0x1C: UIFM signal flags

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6	RW	0	Set to 1 when the UIFM decodes a token successfully (e.g. it passes CRC5, PID check and has matching device address).
5	RW	0	Set to 1 when linestate indicates an SE0 symbol.
4	RW	0	Set to 1 when linestate indicates a K symbol.
3	RW	0	Set to 1 when linestate indicates a J symbol.
2	RW	0	Set to 1 if an incoming datapacket fails the CRC16 check.
1	RW	0	Set to the value of the UTMI_RXACTIVE input signal.
0	RW	0	Set to the value of the UTMI_RXERROR input signal

F.9 UIFM Sticky flags: 0x20

These bits define the sticky-ness of the bits in the UIFM IFM FLAGS register. A 1 means that bit will be sticky (hold its value until a 1 is written to that bitfield), or normal, in which case signal updates to the UIFM IFM FLAGS bits may be over-written by subsequent changes in those signals.

- TDO to pin 13 of the xSYS header

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 ${}^{1}_{out}$, ${}^{0}_{out}$, ${}^{0}_{in}$, and ${}^{1}_{in}$. For example, if you choose to use XL0 for xSCOPE I/O, you need to connect up $XL0^{1}_{out}$, $XL0^{0}_{out}$, $XL0^{0}_{in}$, $XL0^{1}_{in}$ as follows:

- $XL0^{1}_{out}$ (X0D43) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- $XL0^{0}_{out}$ (X0D42) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- $XL0^{0}_{in}$ (X0D41) to pin 14 of the xSYS header.
- $XL0^{1}_{in}$ (X0D40) to pin 18 of the xSYS header.

H.5 Boot

- ☐ The device is connected to a QSPI flash for booting, connected to X0D01, X0D04..X0D07, and X0D10 (Section 8). If not, you must boot the device through OTP or JTAG, or set it to boot from SPI and connect a SPI flash.
- ☐ The Flash that you have chosen is supported by **xflash**, or you have created a specification file for it.

H.6 JTAG, XScope, and debugging

- ☐ You have decided as to whether you need an XSYS header or not (Section G)
- ☐ If you have not included an XSYS header, you have devised a method to program the SPI-flash or OTP (Section G).

H.7 GPIO

- ☐ You have not mapped both inputs and outputs to the same multi-bit port.
- ☐ Pins X0D04, X0D05, X0D06, and X0D07 are output only and are, during and after reset, pulled high and low appropriately (Section 8)
- ☐ Pins X2D04, X2D05, X2D06 and X2D07 are output only and during and after reset, X2D06 is pulled high and X2D04, X2D05, and X2D07 are pulled low (Section 8)

H.8 Multi device designs

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

- ☐ One device is connected to a QSPI or SPI flash for booting.
- ☐ Devices that boot from link have, for example, X0D06 pulled high and have link XL0 connected to a device to boot from (Section 8).

I PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS2-U24A-512-FB374. Each of the following sections contains items to check for each design.

I.1 Ground Plane

☐ Each ground ball has a via to minimize impedance and conduct heat away from the device. (Section 12.4)

☐ Other than ground vias, there are no (or only a few) vias underneath or closely around the device. This create a good, solid, ground plane.

I.2 Power supply decoupling

☐ The decoupling capacitors are all placed close to a supply pin (Section 12).

☐ The decoupling capacitors are spaced around the device (Section 12).

☐ The ground side of each decoupling capacitor has a direct path back to the center ground of the device.

I.3 PLL_AVDD

☐ The PLL_AVDD filter (especially the capacitor) is placed close to the PLL_AVDD pin (Section 12).