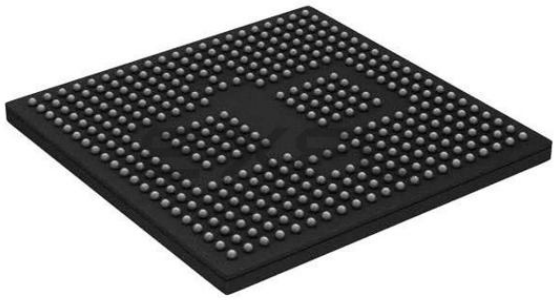


**Welcome to E-XFL.COM**

### 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 | Obsolete |
| Core Processor | XCore |
| Core Size | 32-Bit 32-Core |
| Speed | 4000MIPS |
| Connectivity | USB |
| Peripherals | - |
| Number of I/O | 176 |
| Program Memory Size | 2MB (2M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 1M x 8 |
| Voltage - Supply (Vcc/Vdd) | 0.95V ~ 3.6V |
| Data Converters | - |
| Oscillator Type | External |
| Operating Temperature | 0°C ~ 70°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/xuf232-1024-fb374-c40 |

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
- **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
- **USB** The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 10
- **Flash** The device has a built-in 4MBflash. Section 8
- **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 11

## 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 XUF232-1024-FB374 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - 32 real-time logical cores on 4 xCORE tiles
  - Cores share up to 2000 MIPS
    - Up to 4000 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→64-bit MAC instructions for DSP, arithmetic and user-definable cryptographic functions
- **Dual USB PHY, fully compliant with USB 2.0 specification**
- **Programmable I/O**
  - 176 general-purpose I/O pins, configurable as input or output
    - Up to 56 x 1bit port, 22 x 4bit port, 13 x 8bit port, 6 x 16bit port, 4 x 32bit port
    - 8 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 64 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 1024KB internal single-cycle SRAM (max 256KB per tile) for code and data storage
  - 32KB internal OTP (max 8KB per tile) for application boot code
  - 4MB internal flash for application code and overlays
- **Hardware resources**
  - 24 clock blocks (6 per tile)
  - 40 timers (10 per tile)
  - 16 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**
  - 40: 2000 MIPS
- **Power Consumption**
  - 1140 mA (typical)
- **374-pin FBGA package 0.8 mm pitch**

# 4 Signal Description

This section lists the signals and I/O pins available on the XUF232-1024-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 a weak pull-down or pull-up resistor. On GPIO pins this resistor can be enabled.
- 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 |
| TRST_N | Test reset input | Input | IO, PU, ST |

# 6 Product Overview

The XUF232-1024-FB374 is a powerful device that consists of four 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 8 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 | 7 | 8 |
| 20 | 2000 MIPS | 500 MHz | 100 | 100 | 100 | 100 | 100 | 83 | 71 | 63 |

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 XUF232-1024-FB374, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

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.

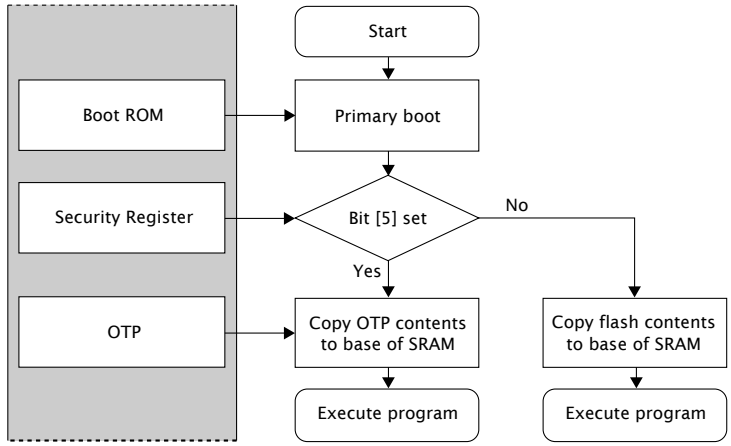



**Figure 9:** Boot procedure

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 802.3); the CRC register is initialized with 0xFFFFFFFF and the residue is inverted to produce the CRC.

### 8.1 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 10 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 data in four sectors each containing 512 rows of 32 bits which can be used to

| 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. |
| Disable Global Debug | 14 | Disables access to the DEBUG_N pin. |
| | 21..15 | General purpose software accessable security register available to end-users. |
| | 31..22 | General purpose user programmable JTAG UserID code extension. |

**Figure 10:** Security register features

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 port with resource ID 0x200100, and the OTP control is on a 16-bit port with ID 0x100300. Programming is performed through `libotp` and `xburn`.

## 9.2 SRAM

Each xCORE Tile integrates a single 256KB SRAM bank for both instructions and data. All internal memory is 32 bits wide, and instructions are either 16-bit or 32-bit. Byte (8-bit), half-word (16-bit) or word (32-bit) accesses are supported and 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.

# 13 DC and Switching Characteristics

## 13.1 Operating Conditions

**Figure 18: Operating conditions**

| Symbol | Parameter | MIN | TYP | MAX | UNITS | Notes |
|---|---|---|---|---|---|---|
| VDD | Tile DC supply voltage | 0.95 | 1.00 | 1.05 | V | |
| VDDIO | I/O supply voltage | 2.30 | 3.30 | 3.60 | V | |
| VDDIOT 3v3 | I/O supply voltage | 3.135 | 3.30 | 3.465 | V | |
| VDDIOT 2v5 | I/O supply voltage | 2.375 | 2.50 | 2.625 | V | |
| VDD33 | Peripheral supply | 3.135 | 3.30 | 3.465 | V | |
| PLL_AVDD | PLL analog supply | 0.95 | 1.00 | 1.05 | V | |
| Cl | xCORE Tile I/O load capacitance | | | 25 | pF | |
| Ta | Ambient operating temperature (Commercial) | 0 | | 70 | ˚C | |
| | Ambient operating temperature (Industrial) | -40 | | 85 | ˚C | |
| Tj | Junction temperature | | | 125 | ˚C | |
| Tstg | Storage temperature | -65 | | 150 | ˚C | |

## 13.2 DC Characteristics

**Figure 19: DC characteristics**

| Symbol | Parameter | MIN | TYP | MAX | UNITS | Notes |
|---|---|---|---|---|---|---|
| V(IH) | Input high voltage | 2.00 | | 3.60 | V | A |
| V(IL) | Input low voltage | -0.30 | | 0.70 | V | A |
| V(OH) | Output high voltage | 2.20 | | | V | B, C |
| V(OL) | Output low voltage | | | 0.40 | V | B, C |
| R(PU) | Pull-up resistance | | 35K | | Ω | D |
| R(PD) | Pull-down resistance | | 35K | | Ω | D |

A All pins except power supply pins.

B All general-purpose I/Os are nominal 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry.

## 13.3 ESD Stress Voltage

**Figure 20: ESD stress voltage**

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

## 13.4 Reset Timing

**Figure 21:** Reset timing

| Symbol | Parameters | MIN | TYP | MAX | UNITS | Notes |
|---|---|---|---|---|---|---|
| T(RST) | Reset pulse width | 5 | | | µs | |
| T(INIT) | Initialization time | | | 150 | µs | A |

A Shows the time taken to start booting after RST_N has gone high.

## 13.5 Power Consumption

**Figure 22:** xCORE Tile currents

| Symbol | Parameter | MIN | TYP | MAX | UNITS | Notes |
|---|---|---|---|---|---|---|
| I(DDCQ) | Quiescent VDD current | | 90 | | mA | A, B, C |
| PD | Tile power dissipation | | 325 | | µW/MIPS | A, D, E, F |
| IDD | Active VDD current | | 1140 | 1400 | mA | A, G |
| I(ADDPLL) | PLL_AVDD current | | 5 | 7 | mA | H |
| I(VDD33) | VDD33 current | | 53.4 | | mA | I |
| I(USB_VDD) | USB_VDD current | | 16.6 | | 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 23:** Clock

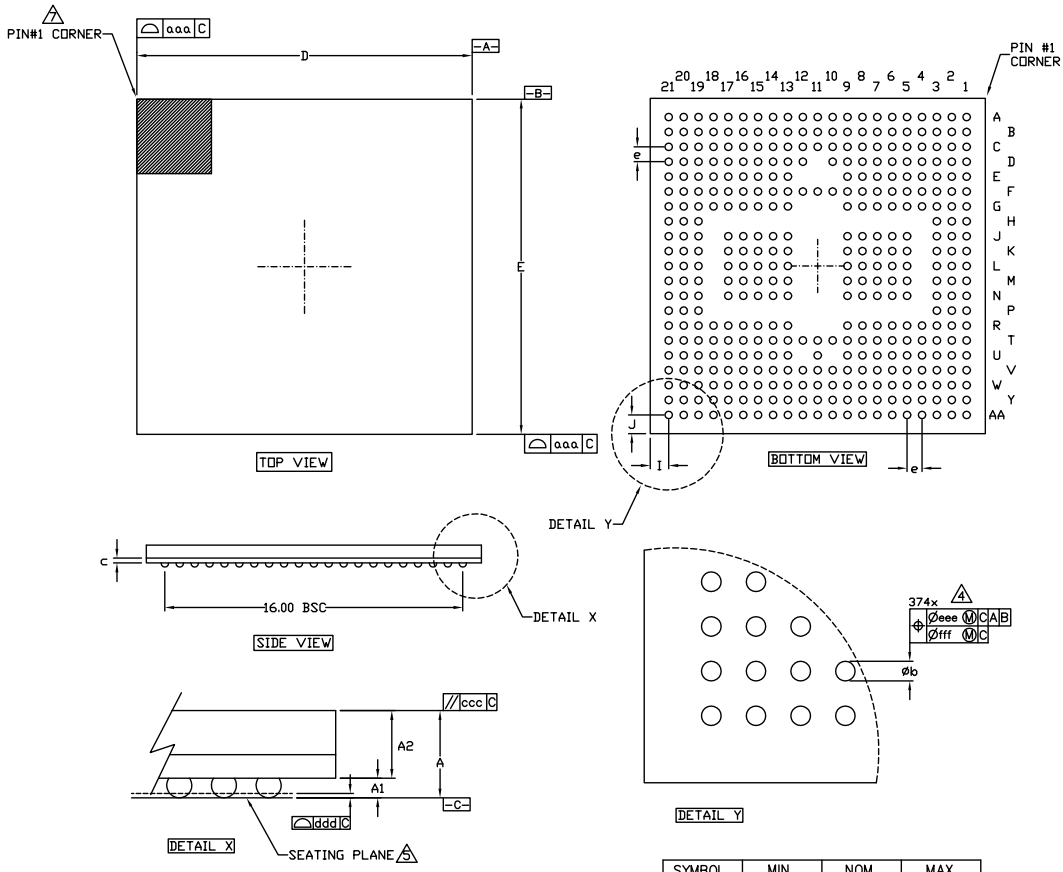
| Symbol | Parameter | MIN | TYP | MAX | UNITS | Notes |
|---|---|---|---|---|---|---|
| f | Frequency | 3.25 | 24 | 100 | MHz | |
| SR | Slew rate | 0.10 | | | V/ns | |
| TJ(LT) | Long term jitter (pk-pk) | | | 2 | % | A |
| f(MAX) | Processor clock frequency | | | 500 | MHz | B |

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,

# 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 XUF232-1024-FB374

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

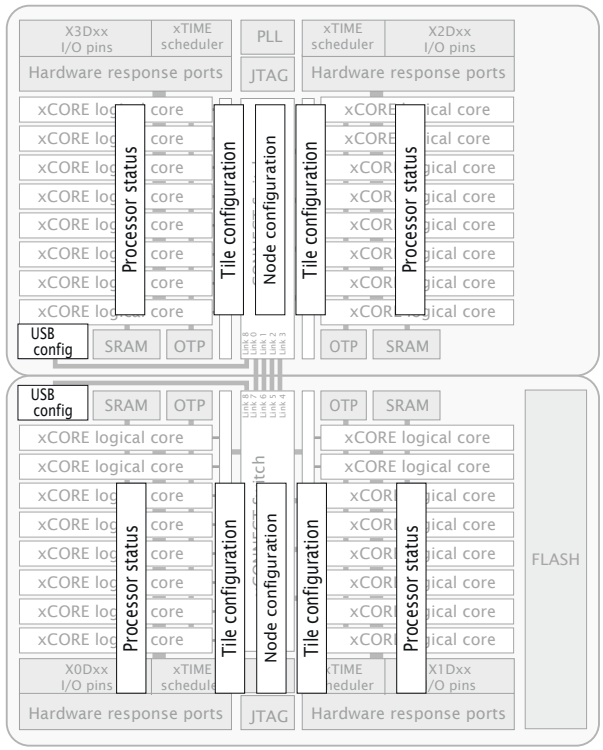



**Figure 29:** Registers

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.

# 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 |

**Figure 30:** Summary

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:26 | RO | - | Reserved |
| 25:18 | RW | 0 | RGMII TX data delay value (in PLL output cycle increments) |
| 17:9 | RW | 0 | RGMII TX clock divider value. TX clk rises when counter (clocked by PLL output) reaches this value and falls when counter reaches (value»1). Value programmed into this field should be actual divide value required minus 1 |
| 8 | RW | 0 | Enable RGMII interface periph ports |
| 7:6 | RO | - | Reserved |
| 5 | RW | 0 | Select the dynamic mode (1) for the clock divider when the clock divider is enabled. In dynamic mode the clock divider is only activated when all active threads are paused. In static mode the clock divider is always enabled. |
| 4 | RW | 0 | Enable the clock divider. This divides the output of the PLL to facilitate one of the low power modes. |
| 3 | RO | - | Reserved |
| 2 | RW | | Select between UTMI (1) and ULPI (0) mode. |
| 1 | RW | | Enable the ULPI Hardware support module |
| 0 | RO | - | Reserved |

**0x02:** xCORE Tile control

## 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

**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

# 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, ...)` 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

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.

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 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 |
| 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.

| 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. |

**0x40 .. 0x47:** PLink status and network

## 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.

| 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 witin a transmit token -1. |
| 10:0 | RW | 0 | Specify min. number of idle system clocks between two continuous transmit tokens -1. |

**0x80 .. 0x88:** Link configuration and initialization

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 |
|---|---|---|---|
| 31:24 | RW | 0 | Bit mask that determines which flags in UIFM_IFM_FLAG[6:0] contribute to port 1?. If any flag listed in this bitmask is high, port 1? will be high. |
| 23:16 | 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. |
| 15:8 | 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. |
| 7:0 | 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

# K Associated Design Documentation

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

# L 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-UF Link Performance and Design Guidelines | Link timings | |
| XS1-UF Clock Frequency Control | Advanced clock control | |