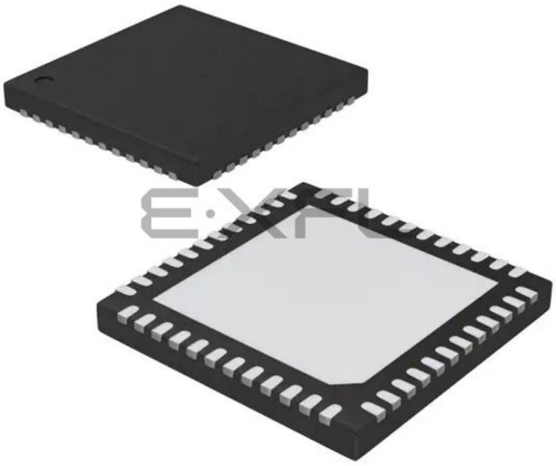
# XMOS - XUF208-256-QF48-I10 Datasheet



| Details | |
|---|---|
| Product Status | Obsolete |
| Core Processor | XCore |
| Core Size | 32-Bit 8-Core |
| Speed | 1000MIPS |
| Connectivity | USB |
| Peripherals | - |
| Number of I/O | 27 |
| Program Memory Size | 1MB (1M x 8) |
| Program Memory Type | FLASH |
| EEPROM Size | - |
| RAM Size | 256K 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 | 48-UFQFN Exposed Pad |
| Supplier Device Package | 48-UQFN (6x6) |
| Purchase URL | https://www.e-xfl.com/product-detail/xmos/xuf208-256-qf48-i10 |

# 5 Example Application Diagram

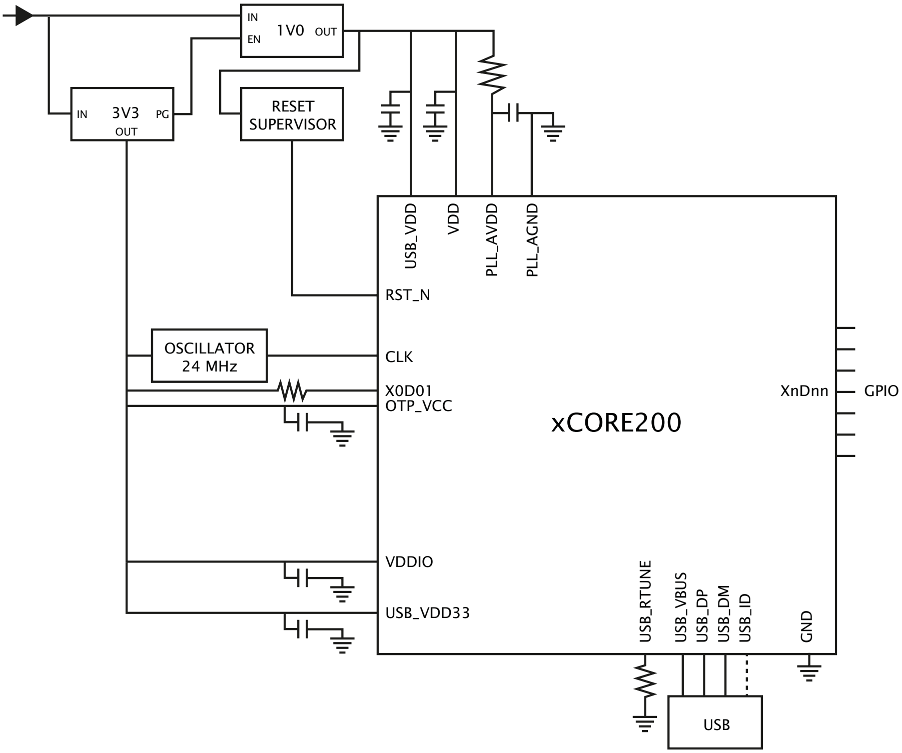


**Figure 2:** Simplified Reference Schematic

- 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.

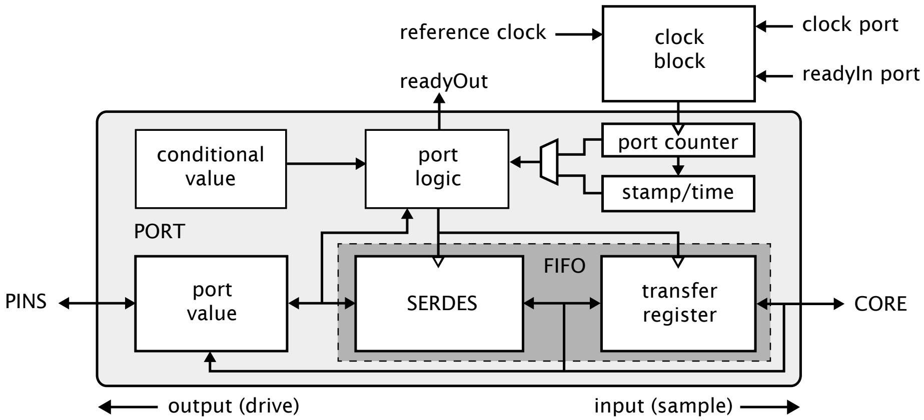


**Figure 4:** Port block diagram

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.

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 $\mu$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.

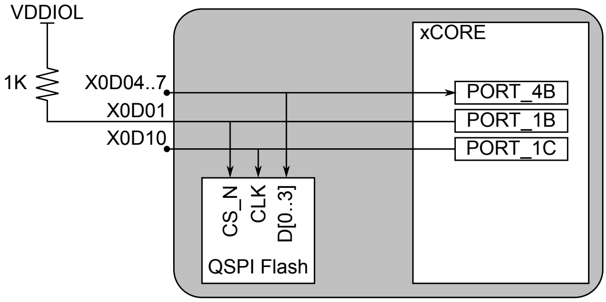


**Figure 8:** QSPI port connectivity

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 accessable security register available to end-users. |
|  | 31..22 | General purpose user programmable JTAG UserID code extension. |

**Figure 10:** Security register features

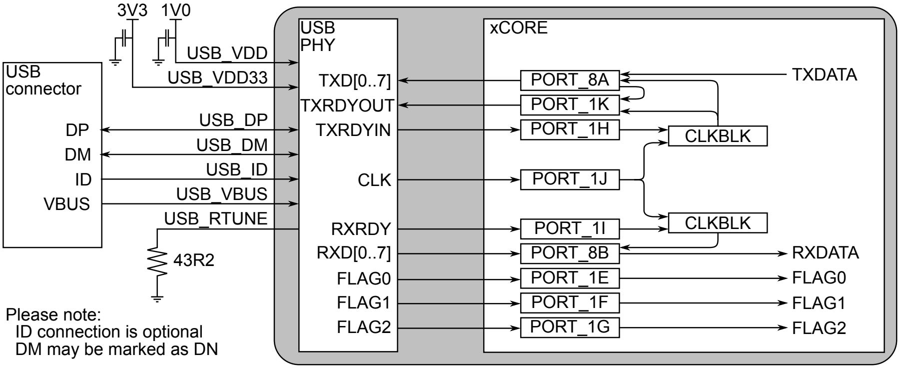
are executed within one tile clock cycle. There is no dedicated external memory interface, although data memory can be expanded through appropriate use of the ports.

## 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, libxud_s.a, 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.

An external resistor of 43.2 ohm (1% tolerance) should connect USB_RTUNE to ground, as close as possible to the device.



Please note:
ID connection is optional
DM may be marked as DN

**Figure 11:** USB port functions

Figure 11 shows how 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.

### 10.1 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.

The JTAG chain structure is illustrated in Figure 12. 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 JTAG module can be reset by holding TMS high for five clock cycles.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 13.

**Figure 13:** IDCODE return value

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

- Route high speed USB traces over continuous power planes, with no breaks. If a trade-off must be made, changing signal layers is preferable to crossing plane splits.
- Follow the $20 \times h$ rule; keep traces $20 \times h$ (the height above the power plane) away from the edge of the power plane.
- Use a minimum of vias in high speed USB traces.
- Avoid corners in the trace. Where necessary, rather than turning through a 90 degree angle, use two 45 degree turns or an arc.
- DO NOT route USB traces near clock sources, clocked circuits or magnetic devices.
- Avoid stubs on high speed USB signals.

### 12.3 Land patterns and solder stencils

The package is a 48 pin Quad Flat No lead package (QFN) with exposed ground paddle/heat slug on a 0.4mm pitch.

The land patterns and solder stencils will depend on the PCB manufacturing process. We recommend you design them with using the IPC specifications *"Generic Requirements for Surface Mount Design and Land Pattern Standards"* IPC-7351B. This standard aims to achieve desired targets of heel, toe and side fillets for solder-joints. The mechanical drawings in Section 14 specify the dimensions and tolerances.

### 12.4 Ground and Thermal Vias

Vias under the heat slug into the ground plane of the PCB are recommended for a low inductance ground connection and good thermal performance. Typical designs could use 9 vias in a 3 x 3 grid, equally spaced across the heat slug.

### 12.5 Moisture Sensitivity

XMOS devices are, like all semiconductor devices, susceptible to moisture absorption. When removed from the sealed packaging, the devices slowly absorb moisture from the surrounding environment. If the level of moisture present in the device is too high during reflow, damage can occur due to the increased internal vapour pressure of moisture. Example damage can include bond wire damage, die lifting, internal or external package cracks and/or delamination.

All XMOS devices are Moisture Sensitivity Level (MSL) 3 - devices have a shelf life of 168 hours between removal from the packaging and reflow, provided they are stored below 30C and 60% RH. If devices have exceeded these values or an included moisture indicator card shows excessive levels of moisture, then the parts should be baked as appropriate before use. This is based on information from *Joint IPC/JEDEC Standard For Moisture/Reflow Sensitivity Classification For Nonhermetic Solid State Surface-Mount Devices* J-STD-020 Revision D.

# 13 DC and Switching Characteristics

## 13.1 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 | 3.135 | 3.30 | 3.465 | V | |
| USB_VDD | USB tile DC supply voltage | 0.95 | 1.00 | 1.05 | 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 | |

**Figure 17:** Operating conditions

## 13.2 DC Characteristics, VDDIO=3V3

| 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 |
| I(PU) | Internal pull-up current (Vin=0V) | -100 | | | µA | D |
| I(PD) | Internal pull-down current (Vin=3.3V) | | | 100 | µA | D |
| I(LC) | Input leakage current | -10 | | 10 | µA | |

**Figure 18:** DC characteristics

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. In order to pull the pin to the opposite state, a 4K7 resistor is recommended to overome the internal pull current.

# 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

**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 indentifier |
| 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.

**0x30 .. 0x33:** Instruction breakpoint address

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:0 | DRW | | Value. |

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

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

**0x40 .. 0x43:** Instruction breakpoint control

| 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. |

## 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

| Bits | Perm | Init | Description |
|---|---|---|---|
| 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

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:0 | DRW | | Value. |

**0x9C .. 0x9F:** Resources 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. |
| 15:2 | RO | - | Reserved |
| 1 | DRW | 0 | When 0 break when condition A is met. When 1 = break when condition B is met. |
| 0 | DRW | 0 | When 1 the instruction breakpoint is enabled. |

**0x00:** Device identification

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:24 | CRO | | Processor ID of this XCore. |
| 23:16 | CRO | | Number of the node in which this XCore is located. |
| 15:8 | CRO | | XCore revision. |
| 7:0 | CRO | | XCore version. |

## C.2 xCORE Tile description 1: 0x01

This register describes the number of logical cores, synchronisers, locks and channel ends available on this xCORE tile.

**0x01:** xCORE Tile description 1

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:24 | CRO | | Number of channel ends. |
| 23:16 | CRO | | Number of the locks. |
| 15:8 | CRO | | Number of synchronisers. |
| 7:0 | RO | - | Reserved |

## C.3 xCORE Tile description 2: 0x02

This register describes the number of timers and clock blocks available on this xCORE tile.

**0x02:** xCORE Tile description 2

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:16 | RO | - | Reserved |
| 15:8 | CRO | | Number of clock blocks. |
| 7:0 | CRO | | Number of timers. |

## C.4 Control PSwitch permissions to debug registers: 0x04

This register can be used to control whether the debug registers (marked with permission CRW) are accessible through the tile configuration registers. When this bit is set, write -access to those registers is disabled, preventing debugging of the xCORE tile over the interconnect.

**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 | 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. |

**0x06:** PLL settings

## 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 |

**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.

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

# E USB Node Configuration

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

**Figure 33:** Summary

| Number | Perm | Description |
|---|---|---|
| 0x00 | RO | Device identification register |
| 0x04 | RW | Node configuration register |
| 0x05 | RW | Node identifier |
| 0x51 | RW | System clock frequency |
| 0x80 | RW | Link Control and Status |

## E.1 Device identification register: 0x00

This register contains version information, and information on power-on behavior.

**0x00:** Device identification register

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:24 | RO | 0x0F | Chip identifier |
| 23:16 | RO | - | Reserved |
| 15:8 | RO | 0x02 | Revision number of the USB block |
| 7:0 | RO | 0x00 | Version number of the USB block |

## E.2 Node configuration register: 0x04

This register is used to set the communication model to use (1 or 3 byte headers), and to prevent any further updates.

**0x04:** Node configuration register

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31 | RW | 0 | Set to 1 to disable further updates to the node configuration and link control and status registers. |
| 30:1 | RO | - | Reserved |
| 0 | RW | 0 | Header mode. 0: 3-byte headers; 1: 1-byte headers. |

**0x10:** UIFM on-the-go control

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:8 | RO | - | Reserved |
| 7 | RW | 0 | Set to 1 to switch UIFM to EXTVBUSIND mode. |
| 6 | RW | 0 | Set to 1 to switch UIFM to DRVVBUSEXT mode. |
| 5 | RO | - | Reserved |
| 4 | RW | 0 | Set to 1 to switch UIFM to UTMI+ CHRGVBUS mode. |
| 3 | RW | 0 | Set to 1 to switch UIFM to UTMI+ DISCHRGVBUS mode. |
| 2 | RW | 0 | Set to 1 to switch UIFM to UTMI+ DMPULLDOWN mode. |
| 1 | RW | 0 | Set to 1 to switch UIFM to UTMI+ DPPULLDOWN mode. |
| 0 | RW | 0 | Set to 1 to switch UIFM to IDPULLUP mode. |

## F.6 UIFM on-the-go flags: 0x14

Status flags used for on-the-go negotiation

**0x14:** UIFM on-the-go flags

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:6 | RO | - | Reserved |
| 5 | RO | 0 | Value of UTMI+ Bvalid flag. |
| 4 | RO | 0 | Value of UTMI+ IDGND flag. |
| 3 | RO | 0 | Value of UTMI+ HOSTDIS flag. |
| 2 | RO | 0 | Value of UTMI+ VBUSVLD flag. |
| 1 | RO | 0 | Value of UTMI+ SESSVLD flag. |
| 0 | RO | 0 | Value of UTMI+ SESSEND flag. |

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.

# I PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS2-UF8A-256-QF48. Each of the following sections contains items to check for each design.

## I.1 Ground Plane

☐ Multiple vias (eg, 9) have been used to connect the center pad to the PCB ground plane. These 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).