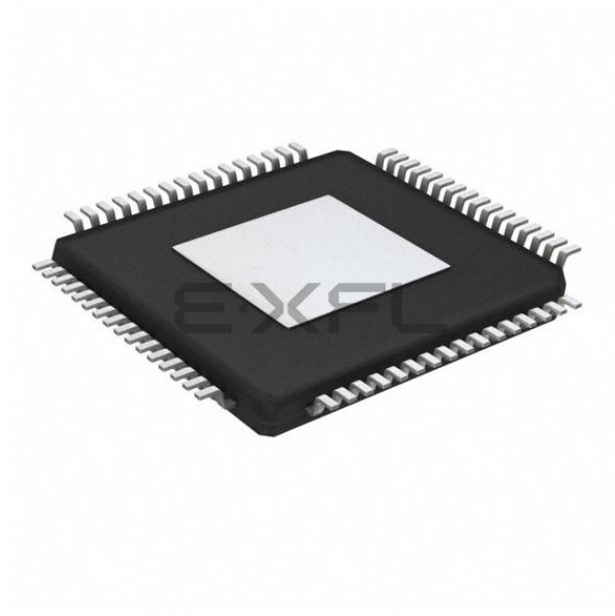


**Welcome to E-XFL.COM**

## 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 8-Core |
| Speed | 1000MIPS |
| Connectivity | USB |
| Peripherals | - |
| Number of I/O | 33 |
| Program Memory Size | - |
| Program Memory Type | ROMless |
| 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 | 64-TQFP Exposed Pad |
| Supplier Device Package | 64-TQFP (10x10) |
| Purchase URL | https://www.e-xfl.com/product-detail/xmos/xu208-256-tq64-i10 |

# 2 XU208-256-TQ64 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Eight real-time logical cores
  - Core share up to 500 MIPS
    - Up to 1000 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
- **USB PHY, fully compliant with USB 2.0 specification**
- **Programmable I/O**
  - 33 general-purpose I/O pins, configurable as input or output
    - Up to 9 x 1bit port, 5 x 4bit port, 3 x 8bit port, 1 x 16bit port
    - 1 xCONNECT link
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 32 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 256KB internal single-cycle SRAM for code and data storage
  - 8KB internal OTP for application boot code
- **Hardware resources**
  - 6 clock blocks
  - 10 timers
  - 4 locks
- **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**
  - 10: 500 MIPS
- **Power Consumption**
  - 170 mA (typical)
- **64-pin TQFP package 0.5 mm pitch**

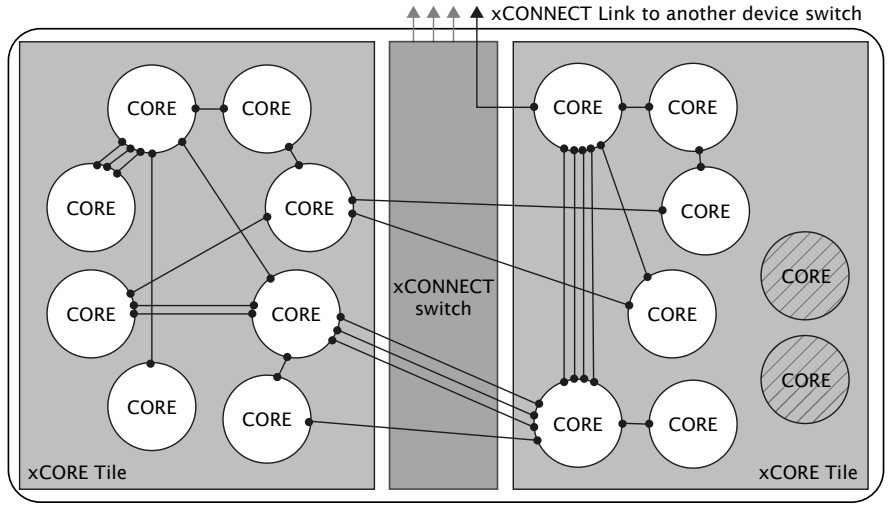



**Figure 6:** Switch, links and channel ends

and packet switched data can both be supported efficiently. Streams provide the fastest possible data rates between xCORE Tiles (up to 250 MBit/s), but each stream requires a single link to be reserved between switches on two tiles. All packet communications can be multiplexed onto a single link.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-U Link Performance and Design Guide, X2999.

# 7 PLL

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

**Figure 7:** The initial PLL multiplier values

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

Figure 7 also lists the values of $OD$, $F$ and $R$, which are the registers that define the ratio of the tile frequency to the oscillator frequency:

$$F_{core} = F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \times \frac{1}{OD+1}$$

$OD$, $F$ and $R$ must be chosen so that $0 \leq R \leq 63$, $0 \leq F \leq 4095$, $0 \leq OD \leq 7$, and $260MHz \leq F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \leq 1.3GHz$. The $OD$, $F$, and $R$ values can be modified by writing to the digital node PLL configuration register.

# 9 Memory

## 9.1 OTP

The 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 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

The 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.

# 10 USB PHY

The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. The PHY is configured through a set of peripheral registers (Appendix F), and data is communicated through ports on the digital node. A library, XUD, is provided to implement *USB-device* functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 14. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles. Two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.

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

## 10.1 USB VBUS

USB_VBUS need not be connected if the device is wholly powered by USB, and the device is used to implement a *USB-device*.

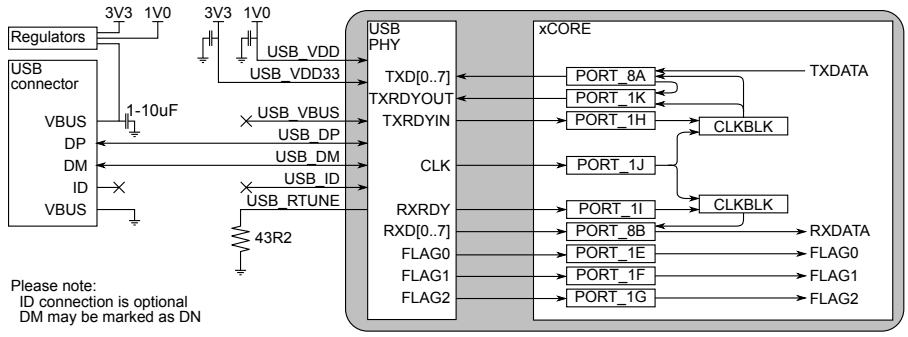



**Figure 14:** Bus powered *USB-device*

If you use the USB PHY to design a self-powered *USB-device*, then the device must be able detect the presence of VBus on the USB connector (so the device can disconnect its pull-up resistors from D+/D- to ensure the device does not have any voltage on the D+/D- pins when VBus is not present, "USB Back Voltage Test"). This requires USB_VBUS to be connected to the VBUS pin of the USB connector as is shown in Figure 15.

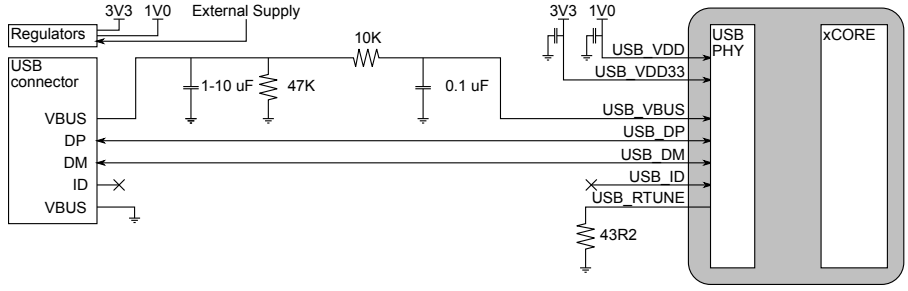



**Figure 15:** Self powered *USB-device*

When connecting a USB cable to the device it is possible an overvoltage transient will be present on VBus due to the inductance of the USB cable combined with the required input capacitor on VBus. The circuit in Figure 15 ensures that the transient does not damage the device. The 10k series resistor and 0.1uF capacitor ensure than any input transient is filtered and does not reach the device. The 47k resistor to ground is a bleeder resistor to discharge the input capacitor when VBus is not present. The 1-10uF input capacitor is required as part of the USB specification. A typical value would be 2.2uF to ensure the 1uF minimum requirement is met even under voltage bias conditions.

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

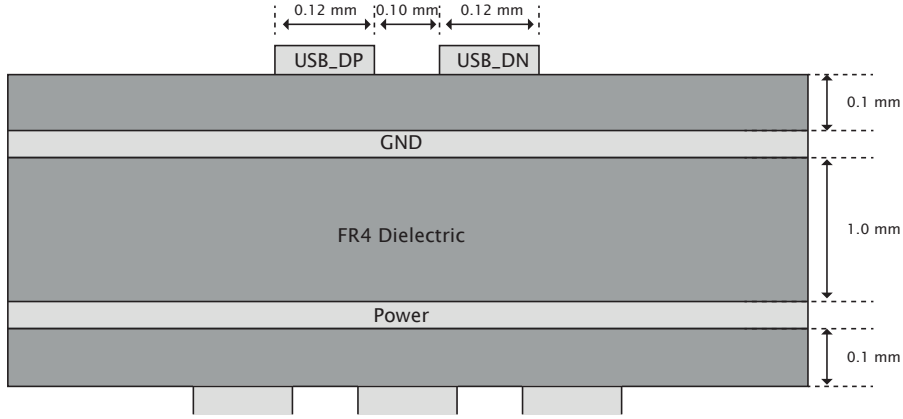
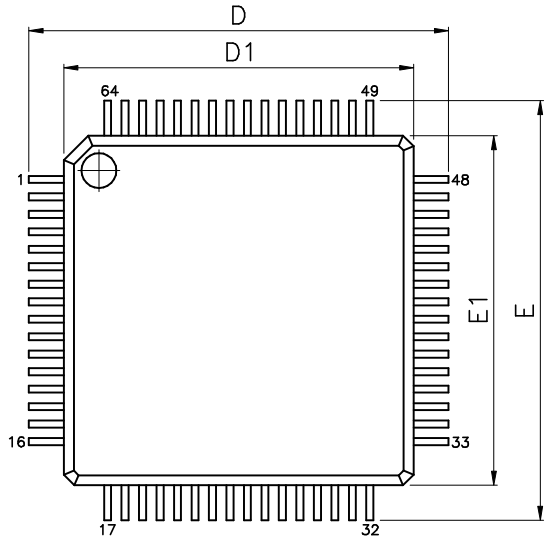


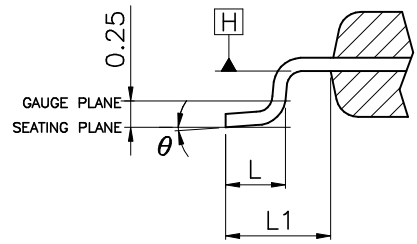

**Figure 20:** Example USB board stack

We recommend that the high-speed clock and high-speed USB differential pairs are routed first before any other routing. When routing high speed USB signals, the following guidelines should be followed:

- High speed differential pairs should be routed together.
- High-speed USB signal pair traces should be trace-length matched. Maximum trace-length mismatch should be no greater than 4mm.
- Ensure that high speed signals (clocks, USB differential pairs) are routed as far away from off-board connectors as possible.
- High-speed clock and periodic signal traces that run parallel should be at least 1.27mm away from USB_DP/USB_DN (see Figure 19).
- Low-speed and non-periodic signal traces that run parallel should be at least 0.5mm away from USB_DP/USB_DN (see Figure 19).
- Route high speed USB signals on the top of the PCB wherever possible.
- 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.

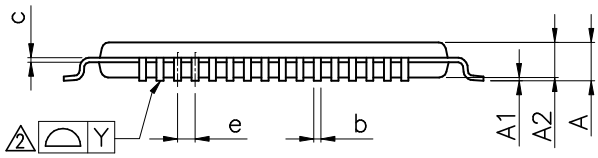
# 14 Package Information

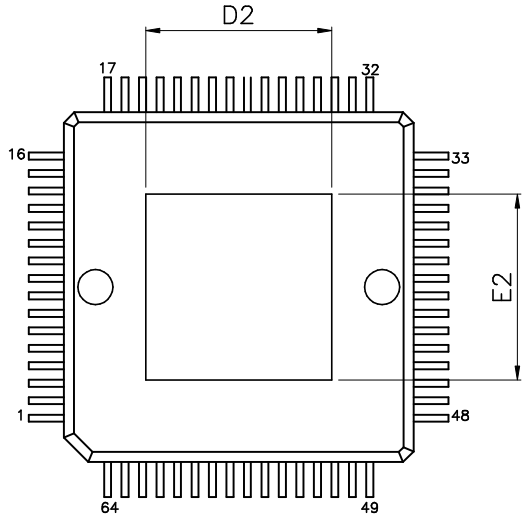



VARIATIONS (ALL DIMENSIONS SHOWN IN MM)

| SYMBOLS | MIN. | NOM. | MAX. |
|---|---|---|---|
| A | – | – | 1.20 |
| A1 | 0.05 | – | 0.15 |
| A2 | 0.95 | 1.00 | 1.05 |
| D | 11.75 | 12.00 | 12.25 |
| D1 | 9.90 | 10.00 | 10.10 |
| 3 D2 | 5.13 | – | 5.48 |
| E | 11.75 | 12.00 | 12.25 |
| E1 | 9.90 | 10.00 | 10.10 |
| 3 E2 | 5.13 | – | 5.48 |
| b | 0.17 | 0.22 | 0.27 |
| c | 0.09 | – | 0.20 |
| L | 0.45 | 0.60 | 0.75 |
| L1 | 1.00 REF | | |
| e | 0.50 BSC | | |
| θ | 0° | 3.5° | 7° |
| Y | 0.08 | | |




NOTES:

1. JEDEC OUTLINE : MS-026 ACD-HD
2. DATUM PLANE H IS LOCATED AT THE BOTTOM OF THE MOLD PARTING LINE COINCIDENT WITH WHERE THE LEAD EXITS THE BODY.
3. DIMENSIONS D1 AND E1 DO NOT INCLUDE MOLD PROTRUSION. ALLOWABLE PROTRUSION IS 0.25 mm PER SIDE. DIMENSIONS D1 AND E1 DO INCLUDE MOLD MISMATCH AND ARE DETERMINED AT DATUM PLANE H .
4. DIMENSION b DOES NOT INCLUDE DAMBAR PROTRUSION.

### 14.1 Part Marking

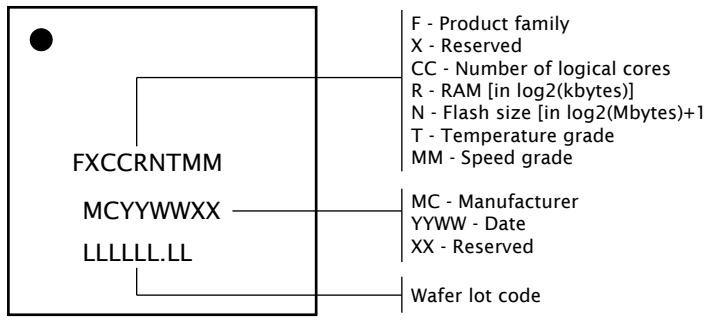



**Figure 31:** Part marking scheme

## 15 Ordering Information

**Figure 32:** Orderable part numbers

| Product Code | Marking | Qualification | Speed Grade |
|---|---|---|---|
| XU208-256-TQ64-C10 | U30880C10 | Commercial | 500 MIPS |
| XU208-256-TQ64-I10 | U30880I10 | Industrial | 500 MIPS |

# 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 34:** Summary

### B.1 RAM base address: 0x00

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

**0x00:** RAM base address

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

### B.2 Vector base address: 0x01

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

**0x01:** Vector base address

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:18 | RW | | The event and interrupt vectors. |
| 17:0 | RO | - | Reserved |

### B.3 xCORE Tile control: 0x02

Register to control features in the xCORE tile

**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

**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. |

**0x41:** PC of logical core 1

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

## C.11 PC of logical core 2: 0x42

Value of the PC of logical core 2.

**0x42:** PC of logical core 2

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

## C.12 PC of logical core 3: 0x43

Value of the PC of logical core 3.

**0x43:** PC of logical core 3

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

## C.13 PC of logical core 4: 0x44

Value of the PC of logical core 4.

**0x44:** PC of logical core 4

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

## C.14 PC of logical core 5: 0x45

Value of the PC of logical core 5.

**0x45:** PC of logical core 5

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

### C.15 PC of logical core 6: 0x46

Value of the PC of logical core 6.

**0x46:** PC of logical core 6

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

### C.16 PC of logical core 7: 0x47

Value of the PC of logical core 7.

**0x47:** PC of logical core 7

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

### C.17 SR of logical core 0: 0x60

Value of the SR of logical core 0

**0x60:** SR of logical core 0

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

### C.18 SR of logical core 1: 0x61

Value of the SR of logical core 1

**0x61:** SR of logical core 1

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

### C.19 SR of logical core 2: 0x62

Value of the SR of logical core 2

**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 |

**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.

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

**0x1F:** Debug source

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

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

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

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

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

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

**0x2C:** UIFM PID

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:4 | RO | - | Reserved |
| 3:0 | RO | 0 | Value of the last received PID. |

### F.13 UIFM Endpoint: 0x30

The last endpoint seen

**0x30:** UIFM Endpoint

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:5 | RO | - | Reserved |
| 4 | RO | 0 | 1 if endpoint contains a valid value. |
| 3:0 | RO | 0 | A copy of the last received endpoint. |

### F.14 UIFM Endpoint match: 0x34

This register can be used to mark UIFM endpoints as special.

**0x34:** UIFM Endpoint match

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:16 | RO | - | Reserved |
| 15:0 | RW | 0 | This register contains a bit for each endpoint. If its bit is set, the endpoint will be supplied on the RX port when ORed with 0x10. |

### F.15 OTG Flags mask: 0x38

**0x38:** OTG Flags mask

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:0 | RW | 0 | Data |

### F.16 UIFM power signalling: 0x3C

**0x3C:** UIFM power signalling

| Bits | Perm | Init | Description |
|---|---|---|---|
| 31:9 | RO | - | Reserved |
| 8 | RW | 0 | Valid |
| 7:0 | RW | 0 | Data |

# G JTAG, xSCOPE and Debugging

If you intend to design a board that can be used with the XMOS toolchain and xTAG debugger, you will need an xSYS header on your board. Figure 39 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

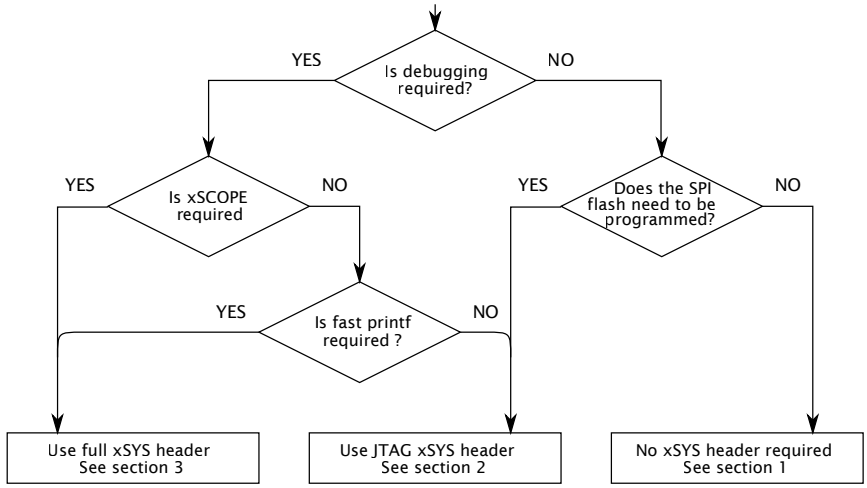



**Figure 39:** Decision diagram for the xSYS header

## G.1 No xSYS header

The use of an xSYS header is optional, and may not be required for volume production designs. However, the XMOS toolchain expects the xSYS header; if you do not have an xSYS header then you must provide your own method for writing to flash/OTP and for debugging.

## G.2 JTAG-only xSYS header

The xSYS header connects to an xTAG debugger, which has a 20-pin 0.1" female IDC header. The design will hence need a male IDC header. We advise to use a boxed header to guard against incorrect plug-ins. If you use a 90 degree angled header, make sure that pins 2, 4, 6, ..., 20 are along the edge of the PCB.

Connect pins 4, 8, 12, 16, 20 of the xSYS header to ground, and then connect:

- TDI to pin 5 of the xSYS header
- TMS to pin 7 of the xSYS header
- TCK to pin 9 of the xSYS header
- TDO to pin 13 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)

### 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).

# L Revision History

| Date | Description |
|---|---|
| 2015-03-20 | Preliminary release |
| 2015-04-14 | Added RST to pins to be pulled hard, and removed reference to TCK from Errata<br>Removed TRST_N references in packages that have no TRST_N |
| 2015-05-06 | Removed references to DEBUG_N |
| 2015-07-09 | Updated electrical characteristics - Section 13 |
| 2015-08-19 | Added I(USB_VDD) - Section 13<br>Added USB layout guidelines - Section 12 |
| 2015-08-27 | Updated part marking - Section 15 |
| 2016-01-05 | Updated Power Supply and Multi Device Designs in Schematics Checklist - Section H |
| 2016-04-20 | Typical internal pull-up and pull down current diagrams added - Section 13 |
| 2017-02-02 | Updated USB VBUS wiring description with bus-powered usb-device instructions - Section 10 |