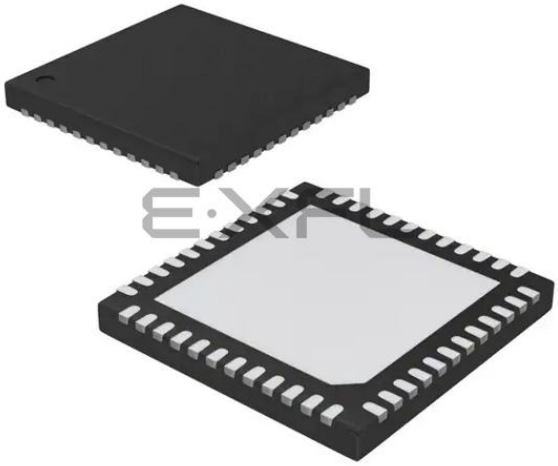

Welcome to E-XFL.COM

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	27
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	0°C ~ 70°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/xu208-256-qf48-c10

1 xCORE Multicore Microcontrollers

The xCORE-200 Series is a comprehensive range of 32-bit multicore microcontrollers that brings the low latency and timing determinism of the xCORE architecture to mainstream embedded applications. Unlike conventional microcontrollers, xCORE multicore microcontrollers execute multiple real-time tasks simultaneously and communicate between tasks using a high speed network. Because xCORE multicore microcontrollers are completely deterministic, you can write software to implement functions that traditionally require dedicated hardware.

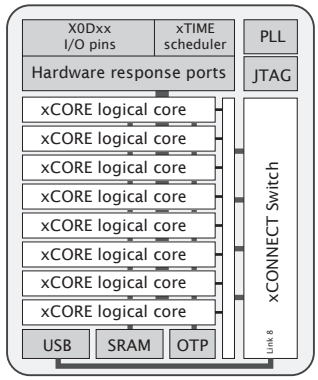

Figure 1: XU208-256-QF48 block diagram

Key features of the XU208-256-QF48 include:

- **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between five and eight 32-bit xCOREs with highly integrated I/O and on-chip memory.
- **Logical cores** Each logical core can execute tasks such as computational code, DSP code, control software (including logic decisions and executing a state machine) or software that handles I/O. Section 6.1
- **xTIME scheduler** The xTIME scheduler performs functions similar to an RTOS, in hardware. It services and synchronizes events in a core, so there is no requirement for interrupt handler routines. The xTIME scheduler triggers cores on events generated by hardware resources such as the I/O pins, communication channels and timers. Once triggered, a core runs independently and concurrently to other cores, until it pauses to wait for more events. Section 6.2
- **Channels and channel ends** Tasks running on logical cores communicate using channels formed between two channel ends. Data can be passed synchronously or asynchronously between the channel ends assigned to the communicating tasks. Section 6.5
- **xCONNECT Switch and Links** Between tiles, channel communications are implemented over a high performance network of xCONNECT Links and routed through a hardware xCONNECT Switch. Section 6.6

Signal	Function	Type	Properties
X0D15	$4C^{1}$ $8B^{1}$ $16A^{9}$ $32A^{29}$	I/O	IO, PD
X0D16	$4D^{0}$ $8B^{2}$ $16A^{10}$	I/O	IO, PD
X0D17	$4D^{1}$ $8B^{3}$ $16A^{11}$	I/O	IO, PD
X0D18	$4D^{2}$ $8B^{4}$ $16A^{12}$	I/O	IO, PD
X0D19	$4D^{3}$ $8B^{5}$ $16A^{13}$	I/O	IO, PD
X0D26	$4E^{0}$ $8C^{0}$ $16B^{0}$	I/O	IO, PD
X0D27	$4E^{1}$ $8C^{1}$ $16B^{1}$	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
X0D35	$1L^{0}$	I/O	IO, PD
X0D36	$1M^{0}$ $8D^{0}$ $16B^{8}$	I/O	IO, PD
X0D37	$1N^{0}$ $8D^{1}$ $16B^{9}$	I/O	IO, PD
X0D38	$1O^{0}$ $8D^{2}$ $16B^{10}$	I/O	IO, PD
X0D39	$1P^{0}$ $8D^{3}$ $16B^{11}$	I/O	IO, PD
X0D40	$X_0LO1^{I}_{in}$ $8D^{4}$ $16B^{12}$	I/O	IO, PD
X0D41	$X_0LO0^{O}_{in}$ $8D^{5}$ $16B^{13}$	I/O	IO, PD
X0D42	$X_0LO0^{O}_{out}$ $8D^{6}$ $16B^{14}$	I/O	IO, PD
X0D43	$X_0LO1^{I}_{out}$ $8D^{7}$ $16B^{15}$	I/O	IO, PD

usb pins (4)			
Signal	**Function**	**Type**	**Properties**
USB_DM	USB Serial Data Inverted	I/O	
USB_DP	USB Serial Data	I/O	
USB_RTUNE	USB resistor	I/O	
USB_VBUS	USB Power Detect Pin	I/O	

System pins (1)			
Signal	**Function**	**Type**	**Properties**
CLK	PLL reference clock	Input	IO, PD, ST

5 Example Application Diagram

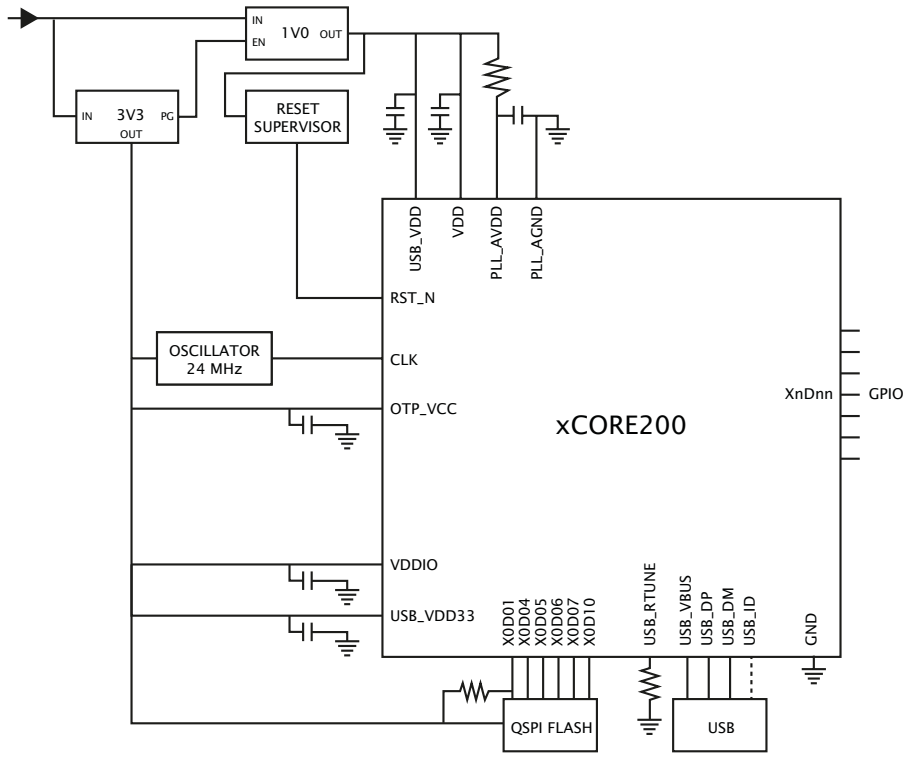

Figure 2: Simplified Reference Schematic

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

6 Product Overview

The XU208-256-QF48 is a powerful device that consists of a single xCORE Tile, which comprises a flexible logical processing cores with tightly integrated I/O and on-chip memory.

6.1 Logical cores

The 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
5	500 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 XU208-256-QF48, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

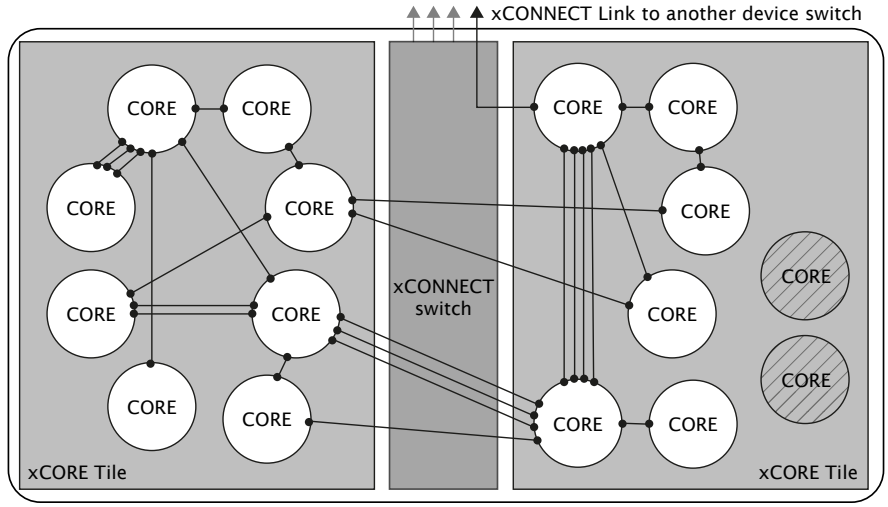

Figure 6: Switch, links and channel ends

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 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.

14.1 Part Marking

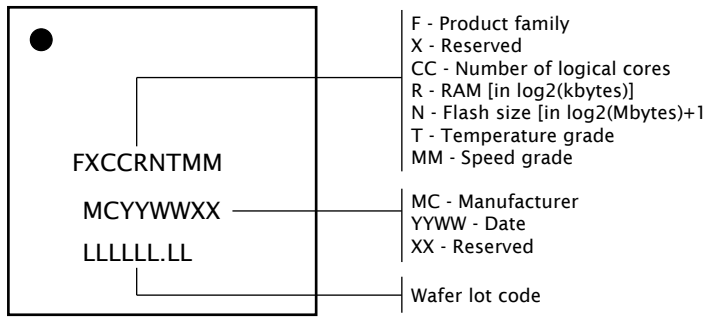

Figure 30: Part marking scheme

15 Ordering Information

Figure 31: Orderable part numbers

Product Code	Marking	Qualification	Speed Grade
XU208-256-QF48-C10	U30880C10	Commercial	500 MIPS
XU208-256-QF48-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 33: Summary

B.5 Security configuration: 0x05

Copy of the security register as read from OTP.

0x05: Security configuration

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

B.6 Ring Oscillator Control: 0x06

There are four free-running oscillators that clock four counters. The oscillators can be started and stopped using this register. The counters should only be read when the ring oscillator has been stopped for at least 10 core clock cycles (this can be achieved by inserting two nop instructions between the SETPS and GETPS). The counter values can be read using four subsequent registers. The ring oscillators are asynchronous to the xCORE tile clock and can be used as a source of random bits.

0x06: Ring Oscillator Control

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Core ring oscillator enable.
0	RW	0	Peripheral ring oscillator enable.

B.7 Ring Oscillator Value: 0x07

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

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.

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.

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.

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.

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

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

0x04: UIFM IFM control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7	RW	0	Set to 1 to enable XEVACKMODE mode.
6	RW	0	Set to 1 to enable SOFISTOKEN mode.
5	RW	0	Set to 1 to enable UIFM power signalling mode.
4	RW	0	Set to 1 to enable IF timing mode.
3	RO	-	Reserved
2	RW	0	Set to 1 to enable UIFM linestate decoder.
1	RW	0	Set to 1 to enable UIFM CHECKTOKENS mode.
0	RW	0	Set to 1 to enable UIFM DOTOKENS mode.

F.3 UIFM Device Address: 0x08

The device address whose packets should be received. 0 until enumeration, it should be set to the assigned value after enumeration.

0x08: UIFM Device Address

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	0	The enumerated USB device address must be stored here. Only packets to this address are passed on.

F.4 UIFM functional control: 0x0C

0x0C: UIFM functional control

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:2	RW	1	Set to 0 to disable UIFM to UTMI+ OPMODE mode.
1	RW	1	Set to 1 to switch UIFM to UTMI+ TERMSELECT mode.
0	RW	1	Set to 1 to switch UIFM to UTMI+ XCVRSELECT mode.

F.5 UIFM on-the-go control: 0x10

This register is used to negotiate an on-the-go connection.

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

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 38 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

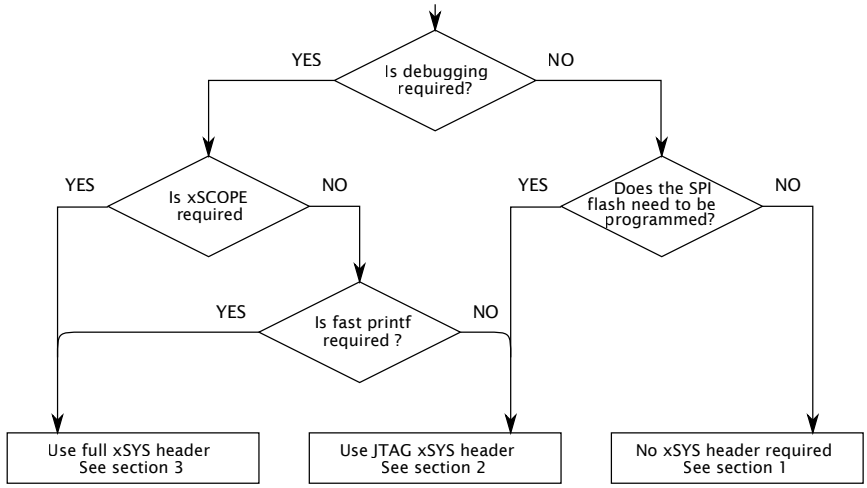

Figure 38: Decision diagram for the xSYS header

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 Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XU208-256-QF48. Each of the following sections contains items to check for each design.

H.1 Power supplies

☐ VDDIO and OTP_VCC supply is within specification before the VDD (core) supply is turned on. Specifically, the VDDIO and OTP_VCC supply is within specification before VDD (core) reaches 0.4V (Section 12).

☐ The VDD (core) supply ramps monotonically (rises constantly) from 0V to its final value (0.95V - 1.05V) within 10ms (Section 12).

☐ The VDD (core) supply is capable of supplying 375 mA (Section 12 and Figure 21).

☐ PLL_AVDD is filtered with a low pass filter, for example an RC filter, see Section 12

H.2 Power supply decoupling

☐ The design has multiple decoupling capacitors per supply, for example at least four0402 or 0603 size surface mount capacitors of 100nF in value, per supply (Section 12).

☐ A bulk decoupling capacitor of at least 10uF is placed on each supply (Section 12).

H.3 Power on reset

☐ The RST_N pins are asserted (low) during or after power up. The device is not used until these resets have taken place.

H.4 Clock

☐ The CLK input pin is supplied with a clock with monotonic rising edges and low jitter.

☐ You have chosen an input clock frequency that is supported by the device (Section 7).

H.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).