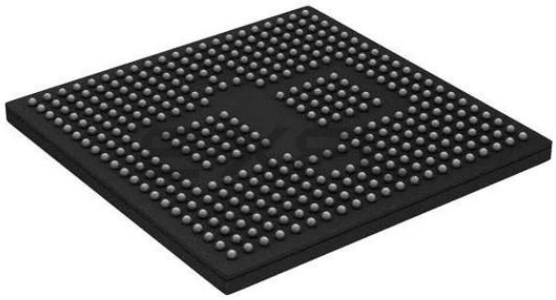

Welcome to E-XFL.COM

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	Discontinued at Digi-Key
Core Processor	XCore
Core Size	32-Bit 32-Core
Speed	4000MIPS
Connectivity	RGMII, USB
Peripherals	-
Number of I/O	176
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	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/xe232-512-fb374-c40

Signal	Function			Type	Properties
X3D42	tx1 (rgmii)	8D[6]	16B[14]	I/O	IOT, PD
X3D43	tx0 (rgmii)	8D[7]	16B[15]	I/O	IOT, PD

System pins (4)			
Signal	**Function**	**Type**	**Properties**
CLK	PLL reference clock	Input	IOL, PD, ST
GLOBAL_DEBUG	Multi-chip debug	I/O	IOL, PU
MODE0	Boot mode select	Input	PU
MODE1	Boot mode select	Input	PU

usb pins (10)			
Signal	**Function**	**Type**	**Properties**
USB_DM_0		I/O	
USB_DM_1		I/O	
USB_DP_0		I/O	
USB_DP_1		I/O	
USB_ID_0		I/O	
USB_ID_1		I/O	
USB_RTUNE_0		I/O	
USB_RTUNE_1		I/O	
USB_VBUS_0		I/O	
VUSB_BUS_1		I/O	

5 Example Application Diagram

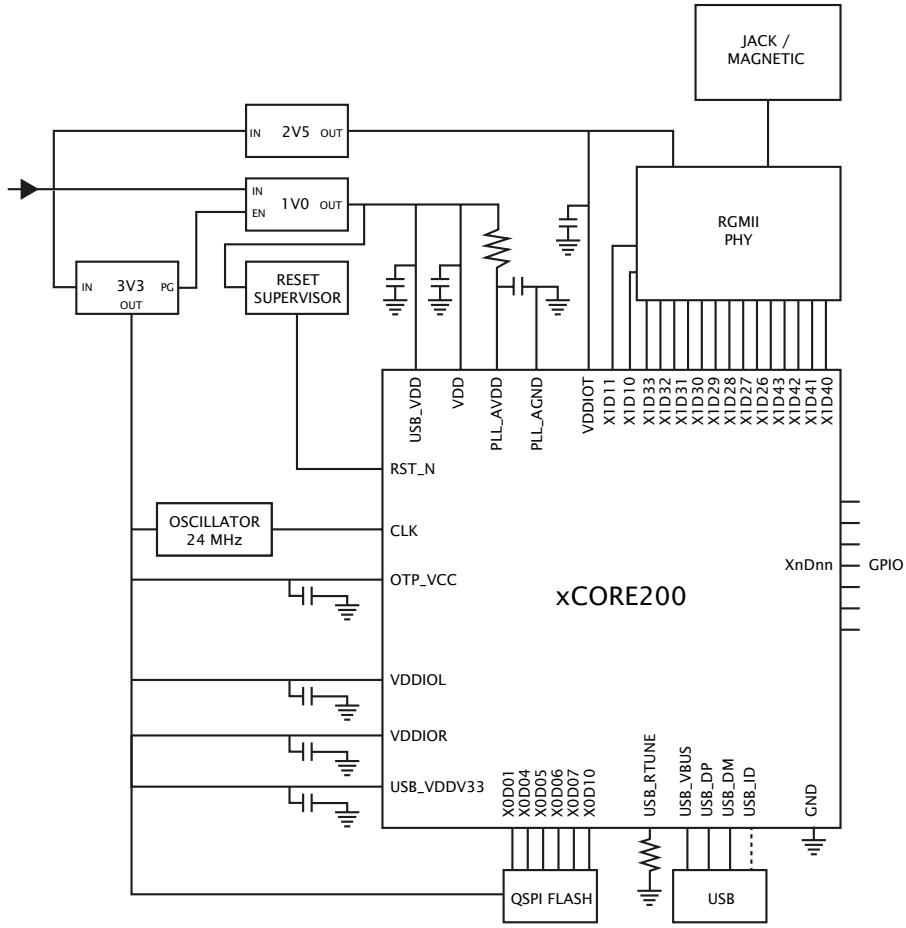

Figure 2: Simplified Reference Schematic

6 Product Overview

The XE232-512-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

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 XE232-512-FB374, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit

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 13: Security register features

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 128KB 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),

The SMI interface should be connected to two one-bit ports that are configured as open-drain IOs, using external pull-ups to 2.5V. Ports 1C and 1D on Tile 1 are notionally allocated for this, but any GPIO can be used for this purpose.

The bundles of RX and TX pins should be wired using matched trace-lengths over an uninterrupted ground-plane. The RGMII pins are supplied through the VDDIOT supply pins, which should be provided with 2.5V. Decouplers should be placed with a short path to VDDIOT and ground. If the PHY supports a 3.3V IO voltage, then a 3.3V supply can be used for VDDIOT.

The RGMII PHY should be configured so that RX_CLK is low during reset of the xCORE. This can be achieved by putting a pull-down resistor on the reset of the PHY, keeping the PHY in reset until the RGMII layer on the xCORE takes the PHY out of reset.

12 JTAG

The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory.

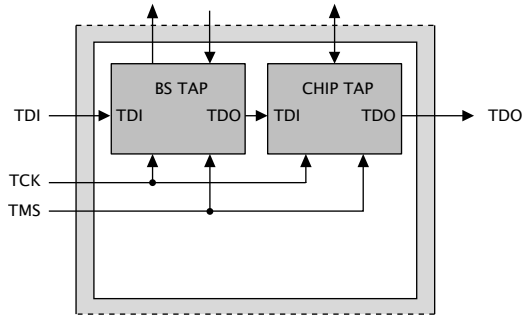

Figure 17: JTAG chain structure

The JTAG chain structure is illustrated in Figure 17. Directly after reset, two TAP controllers are present in the JTAG chain for each xCORE Tile: the boundary scan TAP and the chip TAP. The boundary scan TAP is a standard 1149.1 compliant TAP that can be used for boundary scan of the I/O pins. The chip TAP provides access into the xCORE Tile, switch and OTP 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 18.

Figure 18: 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			

14 DC and Switching Characteristics

14.1 Operating Conditions

Figure 22: 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	

14.2 DC Characteristics

Figure 23: 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.

14.3 ESD Stress Voltage

Figure 24: 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	

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.

B.18 Debug interrupt data: 0x16

On a data watchpoint, this register contains the effective address of the memory operation that triggered the debugger. On a resource watchpoint, it countains the resource identifier.

0x16: Debug interrupt data

Bits	Perm	Init	Description
31:0	DRW		Value.

B.19 Debug core control: 0x18

This register enables the debugger to temporarily disable logical cores. When returning from the debug interrupts, the cores set in this register will not execute. This enables single stepping to be implemented.

0x18: Debug core control

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		1-hot vector defining which threads are stopped when not in debug mode. Every bit which is set prevents the respective thread from running.

B.20 Debug scratch: 0x20 .. 0x27

A set of registers used by the debug ROM to communicate with an external debugger, for example over JTAG. This is the same set of registers as the Debug Scratch registers in the xCORE tile configuration.

0x20 .. 0x27: Debug scratch

Bits	Perm	Init	Description
31:0	DRW		Value.

B.21 Instruction breakpoint address: 0x30 .. 0x33

This register contains the address of the instruction breakpoint. If the PC matches this address, then a debug interrupt will be taken. There are four instruction breakpoints that are controlled individually.

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.

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 35: Summary

C.1 Device identification: 0x00

This register identifies the xCORE Tile

C.24 SR of logical core 7: 0x67

Value of the SR of logical core 7

0x67: SR of logical core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

D.8 System JTAG device ID register: 0x09

0x09: System JTAG device ID register

Bits	Perm	Init	Description
31:28	RO		
27:12	RO		
11:1	RO		
0	RO		

D.9 System USERCODE register: 0x0A

0x0A: System USERCODE register

Bits	Perm	Init	Description
31:18	RO		JTAG USERCODE value programmed into OTP SR
17:0	RO		metal fixable ID code

D.10 Directions 0-7: 0x0C

This register contains eight directions, for packets with a mismatch in bits 7..0 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

0x0C: Directions 0-7

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose dimension is 7.
27:24	RW	0	The direction for packets whose dimension is 6.
23:20	RW	0	The direction for packets whose dimension is 5.
19:16	RW	0	The direction for packets whose dimension is 4.
15:12	RW	0	The direction for packets whose dimension is 3.
11:8	RW	0	The direction for packets whose dimension is 2.
7:4	RW	0	The direction for packets whose dimension is 1.
3:0	RW	0	The direction for packets whose dimension is 0.

D.11 Directions 8-15: 0x0D

This register contains eight directions, for packets with a mismatch in bits 15..8 of the node-identifier. The direction in which a packet will be routed is goverened by the most significant mismatching bit.

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.

0x0D: Directions 8-15

D.12 Reserved: 0x10

Reserved.

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Reserved.
0	RW	0	Reserved.

0x10: Reserved

D.13 Reserved.: 0x11

Reserved.

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Reserved.
0	RW	0	Reserved.

0x11: Reserved.

D.14 Debug source: 0x1F

Contains the source of the most recent debug event.

E.3 Node identifier: 0x05

0x05: Node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	16-bit node identifier. This does not need to be set, and is present for compatibility with XS1-switches.

E.4 System clock frequency: 0x51

0x51: System clock frequency

Bits	Perm	Init	Description
31:7	RO	-	Reserved
6:0	RW	25	Oscillator clock frequency in MHz rounded up to the nearest integer value. Only values between 5 and 100 MHz are valid - writes outside this range are ignored and will be NACKed. This field must be set on start up of the device and any time that the input oscillator clock frequency is changed. It must contain the system clock frequency in MHz rounded up to the nearest integer value.

E.5 Link Control and Status: 0x80

0x80: Link Control and Status

Bits	Perm	Init	Description
31:28	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	1	Specify min. number of idle system clocks between two continuous symbols witin a transmit token -1.
10:0	RW	1	Specify min. number of idle system clocks between two continuous transmit tokens -1.

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

F.17 UIFM PHY control: 0x40

0x40: UIFM PHY control

Bits	Perm	Init	Description
31:19	RO	-	Reserved
18	RW	0	Set to 1 to disable pulldowns on ports 8A and 8B.
17:14	RO	-	Reserved
13	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for reset (se0). Set to 0 to clear.
12	RW	0	After an auto-resume, this bit is set to indicate that the resume signalling was for resume (K). Set to 0 to clear.
11:8	RW	0	Log-2 number of clocks before any linestate change is propagated.
7	RW	0	Set to 1 to use the suspend controller handle to resume from suspend. Otherwise, the program has to poll the linestate_filt field in phy_teststatus.
6:4	RW	0	Control the the conf1,2,3 input pins of the PHY.
3:0	RO	-	Reserved

G Device Errata

This section describes minor operational differences from the data sheet and recommended workarounds. As device and documentation issues become known, this section will be updated the document revised.

To guarantee a logic low is seen on the pins RST_N, MODE[1:0], TMS, and TDI, the driving circuit should present an impedance of less than 100 Ω to ground. Usually this is not a problem for CMOS drivers driving single inputs. If one or more of these inputs are placed in parallel, however, additional logic buffers may be required to guarantee correct operation.

For static inputs tied high or low, the relevant input pin should be tied directly to GND or VDDIO.

H 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.

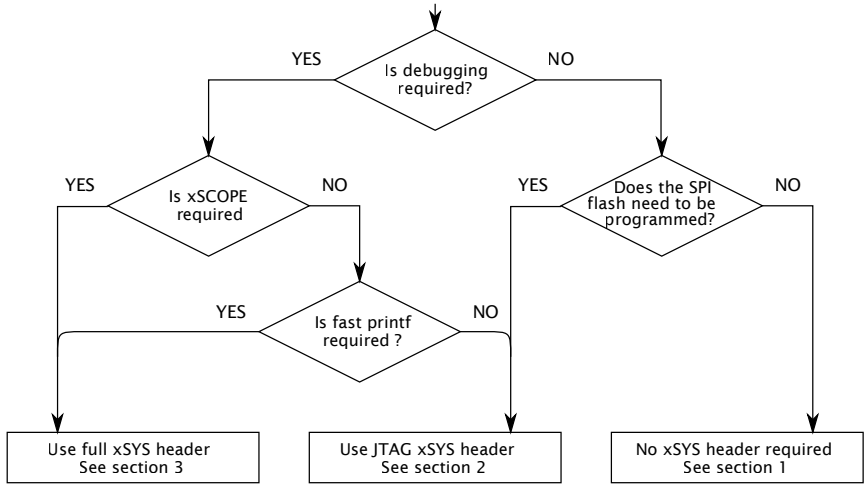

Figure 39: Decision diagram for the xSYS header

H.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.

- ☐ Pins MODE0 and MODE1 are set to the correct value for the chosen oscillator frequency. The MODE settings are shown in the Oscillator section, Section 7. If you have a choice between two values, choose the value with the highest multiplier ratio since that will boot faster.

I.5 RGMII interface

This section can be skipped if you do not have any device connected to the RGMII interface.

- ☐ RX_CLK will be low when the xCORE comes out of reset (see Section 11).
- ☐ VDDIOT has a 2.5V supply.
- ☐ RGMII signals are connected to the appropriate RGMII pins of the xCORE device.

I.6 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.

I.7 JTAG, XScope, and debugging

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

I.8 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, after reset, pulled high and low appropriately (Section 8)

I.9 Multi device designs

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

- ☐ One device is connected to a SPI flash for booting.
- ☐ Devices that boot from link have MODE2 grounded and MODE3 NC. These device must have link XLB connected to a device to boot from (see 8).
- ☐ If you included an XSYS header, you have included buffers for RST_N, TMS, TCK, MODE2, and MODE3 (Section G).