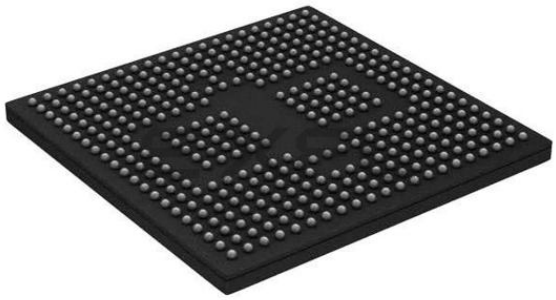

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	XCore
Core Size	32-Bit 24-Core
Speed	4000MIPS
Connectivity	-
Peripherals	-
Number of I/O	176
Program Memory Size	2MB (2M x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	1M x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	374-LFBGA
Supplier Device Package	374-FBGA (18x18)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xlf232-1024-fb374-c40

Table of Contents

Signal	Function					Type	Properties
X1D17	$X_0L3^0_{in}$		4D[1]	8B[3]	16A[11]	I/O	IO, PD
X1D18	$X_0L3^0_{out}$		4D[2]	8B[4]	16A[12]	I/O	IO, PD
X1D19	$X_0L3^1_{out}$		4D[3]	8B[5]	16A[13]	I/O	IO, PD
X1D20			4C[2]	8B[6]	16A[14] 32A[30]	I/O	IO, PD
X1D21			4C[3]	8B[7]	16A[15] 32A[31]	I/O	IO, PD
X1D22	$X_0L3^4_{out}$	1G[0]				I/O	IO, PD
X1D23		1H[0]				I/O	IO, PD
X1D24		1I[0]				I/O	IO, PD
X1D25		1J[0]				I/O	IO, PD
X1D26			4E[0]	8C[0]	16B[0]	I/O	IOT, PD
X1D27			4E[1]	8C[1]	16B[1]	I/O	IOT, PD
X1D28			4F[0]	8C[2]	16B[2]	I/O	IOT, PD
X1D29			4F[1]	8C[3]	16B[3]	I/O	IOT, PD
X1D30			4F[2]	8C[4]	16B[4]	I/O	IOT, PD
X1D31			4F[3]	8C[5]	16B[5]	I/O	IOT, PD
X1D32			4E[2]	8C[6]	16B[6]	I/O	IOT, PD
X1D33			4E[3]	8C[7]	16B[7]	I/O	IOT, PD
X1D34	$X_0L0^2_{out}$	1K[0]				I/O	IO, PD
X1D35	$X_0L0^3_{out}$	1L[0]				I/O	IO, PD
X1D36	$X_0L0^4_{out}$	1M[0]		8D[0]	16B[8]	I/O	IO, PD
X1D37	$X_0L3^4_{in}$	1N[0]		8D[1]	16B[9]	I/O	IO, PD
X1D38	$X_0L3^3_{in}$	1O[0]		8D[2]	16B[10]	I/O	IO, PD
X1D39	$X_0L3^2_{in}$	1P[0]		8D[3]	16B[11]	I/O	IO, PD
X1D40				8D[4]	16B[12]	I/O	IOT, PD
X1D41				8D[5]	16B[13]	I/O	IOT, PD
X1D42				8D[6]	16B[14]	I/O	IOT, PD
X1D43				8D[7]	16B[15]	I/O	IOT, PD
X1D49	$X_0L1^4_{in}$				32A[0]	I/O	IO, PD
X1D50	$X_0L1^3_{in}$				32A[1]	I/O	IO, PD
X1D51	$X_0L1^2_{in}$				32A[2]	I/O	IO, PD
X1D52	$X_0L1^1_{in}$				32A[3]	I/O	IO, PD
X1D53	$X_0L1^0_{in}$				32A[4]	I/O	IO, PD
X1D54	$X_0L1^0_{out}$				32A[5]	I/O	IO, PD
X1D55	$X_0L1^1_{out}$				32A[6]	I/O	IO, PD
X1D56	$X_0L1^2_{out}$				32A[7]	I/O	IO, PD
X1D57	$X_0L1^3_{out}$				32A[8]	I/O	IO, PD
X1D58	$X_0L1^4_{out}$				32A[9]	I/O	IO, PD
X1D61	$X_0L2^4_{in}$				32A[10]	I/O	IO, PD
X1D62	$X_0L2^3_{in}$				32A[11]	I/O	IO, PD
X1D63	$X_0L2^2_{in}$				32A[12]	I/O	IO, PD
X1D64	$X_0L2^1_{in}$				32A[13]	I/O	IO, PD
X1D65	$X_0L2^0_{in}$				32A[14]	I/O	IO, PD
X1D66	$X_0L2^0_{out}$				32A[15]	I/O	IO, PD

(continued)

Signal	Function	Type	Properties
X2D53	$X_2L5^0_{in}$ $32A^4$	I/O	IO, PD
X2D54	$X_2L5^0_{out}$ $32A^5$	I/O	IO, PD
X2D55	$X_2L5^1_{out}$ $32A^6$	I/O	IO, PD
X2D56	$X_2L5^2_{out}$ $32A^7$	I/O	IO, PD
X2D57	$X_2L5^3_{out}$ $32A^8$	I/O	IO, PD
X2D58	$X_2L5^4_{out}$ $32A^9$	I/O	IO, PD
X2D61	$X_2L6^4_{in}$ $32A^{10}$	I/O	IO, PD
X2D62	$X_2L6^3_{in}$ $32A^{11}$	I/O	IO, PD
X2D63	$X_2L6^2_{in}$ $32A^{12}$	I/O	IO, PD
X2D64	$X_2L6^1_{in}$ $32A^{13}$	I/O	IO, PD
X2D65	$X_2L6^0_{in}$ $32A^{14}$	I/O	IO, PD
X2D66	$X_2L6^0_{out}$ $32A^{15}$	I/O	IO, PD
X2D67	$X_2L6^1_{out}$ $32A^{16}$	I/O	IO, PD
X2D68	$X_2L6^2_{out}$ $32A^{17}$	I/O	IO, PD
X2D69	$X_2L6^3_{out}$ $32A^{18}$	I/O	IO, PD
X2D70	$X_2L6^4_{out}$ $32A^{19}$	I/O	IO, PD
X3D00	$X_2L7^2_{in}$ $1A^0$	I/O	IO, PD
X3D01	$X_2L7^1_{in}$ $1B^0$	I/O	IO, PD
X3D02	$X_2L4^0_{in}$ $4A^0$ $8A^0$ $16A^0$ $32A^{20}$	I/O	IO, PD
X3D03	$X_2L4^0_{out}$ $4A^1$ $8A^1$ $16A^1$ $32A^{21}$	I/O	IO, PD
X3D04	$X_2L4^1_{out}$ $4B^0$ $8A^2$ $16A^2$ $32A^{22}$	I/O	IO, PD
X3D05	$X_2L4^2_{out}$ $4B^1$ $8A^3$ $16A^3$ $32A^{23}$	I/O	IO, PD
X3D06	$X_2L4^3_{out}$ $4B^2$ $8A^4$ $16A^4$ $32A^{24}$	I/O	IO, PD
X3D07	$X_2L4^4_{out}$ $4B^3$ $8A^5$ $16A^5$ $32A^{25}$	I/O	IO, PD
X3D08	$X_2L7^4_{in}$ $4A^2$ $8A^6$ $16A^6$ $32A^{26}$	I/O	IO, PD
X3D09	$X_2L7^3_{in}$ $4A^3$ $8A^7$ $16A^7$ $32A^{27}$	I/O	IO, PD
X3D10	$1C^0$	I/O	IOT, PD
X3D11	$1D^0$	I/O	IOT, PD
X3D12	$1E^0$	I/O	IO, PD
X3D13	$1F^0$	I/O	IO, PD
X3D14	$4C^0$ $8B^0$ $16A^8$ $32A^{28}$	I/O	IO, PD
X3D15	$4C^1$ $8B^1$ $16A^9$ $32A^{29}$	I/O	IO, PD
X3D20	$4C^2$ $8B^6$ $16A^{14}$ $32A^{30}$	I/O	IO, PD
X3D21	$4C^3$ $8B^7$ $16A^{15}$ $32A^{31}$	I/O	IO, PD
X3D23	$1H^0$	I/O	IO, PD
X3D24	$1I^0$	I/O	IO, PD
X3D25	$1J^0$	I/O	IO, PD
X3D26	$4E^0$ $8C^0$ $16B^0$	I/O	IOT, PD
X3D27	$4E^1$ $8C^1$ $16B^1$	I/O	IOT, PD
X3D28	$4F^0$ $8C^2$ $16B^2$	I/O	IOT, PD
X3D29	$4F^1$ $8C^3$ $16B^3$	I/O	IOT, PD
X3D30	$4F^2$ $8C^4$ $16B^4$	I/O	IOT, PD
X3D31	$4F^3$ $8C^5$ $16B^5$	I/O	IOT, PD

(continued)

Signal	Function	Type	Properties
X3D32	$4E^{2}$ $8C^{6}$ $16B^{6}$	I/O	IOT, PD
X3D33	$4E^{3}$ $8C^{7}$ $16B^{7}$	I/O	IOT, PD
X3D40	$8D^{4}$ $16B^{12}$	I/O	IOT, PD
X3D41	$8D^{5}$ $16B^{13}$	I/O	IOT, PD
X3D42	$8D^{6}$ $16B^{14}$	I/O	IOT, PD
X3D43	$8D^{7}$ $16B^{15}$	I/O	IOT, PD

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

5 Example Application Diagram

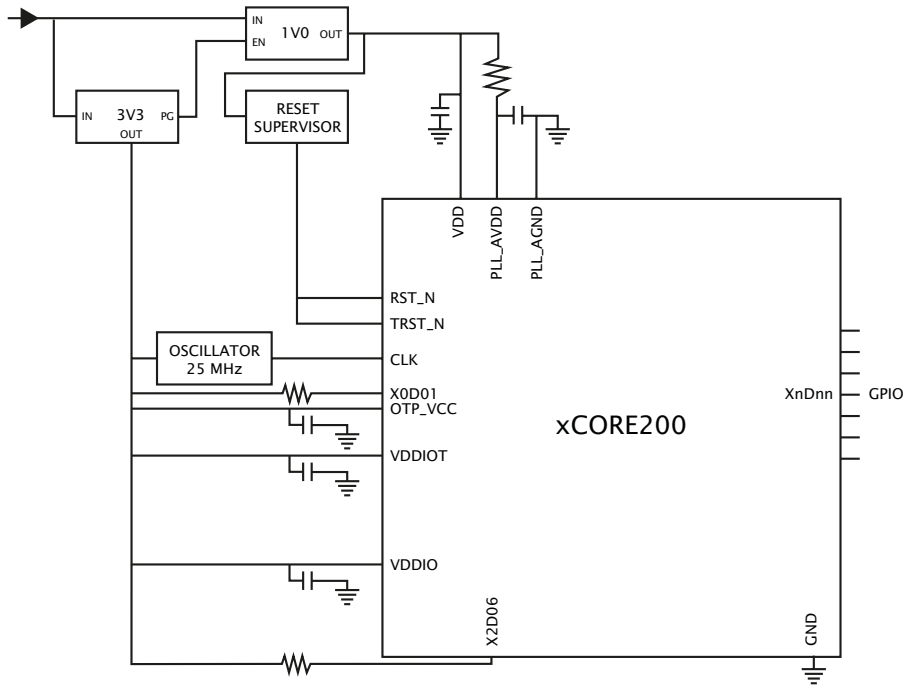

Figure 2: Simplified Reference Schematic

- see Section 11 for details on the power supplies and PCB design

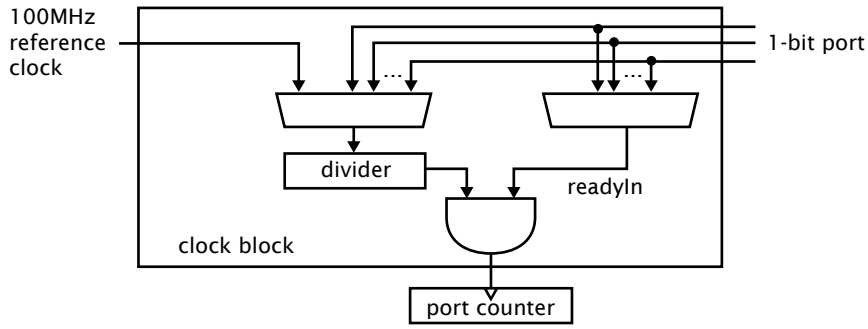

Figure 5: Clock block diagram

A clock block can use a 1-bit port as its clock source allowing external application clocks to be used to drive the input and output interfaces. xCORE-200 clock blocks optionally divide the clock input from a 1-bit port.

In many cases I/O signals are accompanied by strobing signals. The xCORE ports can input and interpret strobe (known as readyIn and readyOut) signals generated by external sources, and ports can generate strobe signals to accompany output data.

On reset, each port is connected to clock block 0, which runs from the xCORE Tile reference clock.

6.5 Channels and Channel Ends

Logical cores communicate using point-to-point connections, formed between two channel ends. A channel-end is a resource on an xCORE tile, that is allocated by the program. Each channel-end has a unique system-wide identifier that comprises a unique number and their tile identifier. Data is transmitted to a channel-end by an output-instruction; and the other side executes an input-instruction. Data can be passed synchronously or asynchronously between the channel ends.

6.6 xCONNECT Switch and Links

XMOS devices provide a scalable architecture, where multiple xCORE devices can be connected together to form one system. Each xCORE device has an xCONNECT interconnect that provides a communication infrastructure for all tasks that run on the various xCORE tiles on the system.

The interconnect relies on a collection of switches and XMOS links. Each xCORE device has an on-chip switch that can set up circuits or route data. The switches are connected by xConnect Links. An XMOS link provides a physical connection between two switches. The switch has a routing algorithm that supports many different topologies, including lines, meshes, trees, and hypercubes.

The links operate in either 2 wires per direction or 5 wires per direction mode, depending on the amount of bandwidth required. Circuit switched, streaming

10 JTAG

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

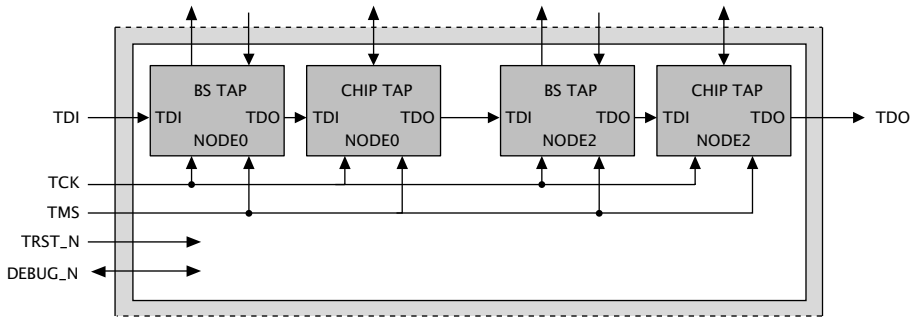

Figure 11: JTAG chain structure

The JTAG chain structure is illustrated in Figure 11. 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 TRST_N pin must be asserted low during and after power up for 100 ns. If JTAG is not required, the TRST_N pin can be tied to ground to hold the JTAG module in reset.

The DEBUG_N pin is used to synchronize the debugging of multiple xCORE Tiles. This pin can operate in both output and input mode. In output mode and when configured to do so, DEBUG_N is driven low by the device when the processor hits a debug break point. Prior to this point the pin will be tri-stated. In input mode and when configured to do so, driving this pin low will put the xCORE Tile into debug mode. Software can set the behavior of the xCORE Tile based on this pin. This pin should have an external pull up of 4K7-47KΩ or left not connected in single core applications.

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 12.

Figure 12: 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			

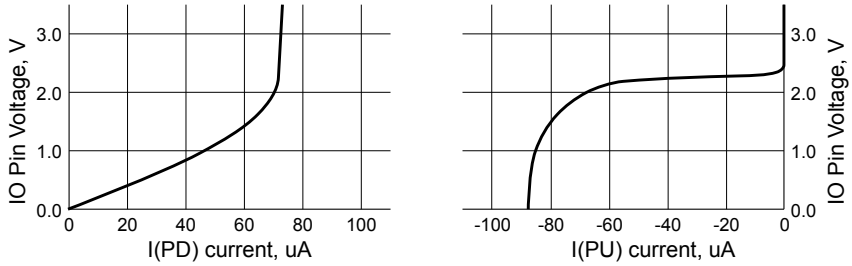

Figure 16: Typical internal pull-down and pull-up currents

12.3 ESD Stress Voltage

Figure 17: 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	

12.4 Reset Timing

Figure 18: Reset timing

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

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

12.5 Power Consumption

Figure 19: xCORE Tile currents

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
I(DDCQ)	Quiescent VDD current		45		mA	A, B, C
PD	Tile power dissipation		325		µW/MIPS	A, D, E, F
IDD	Active VDD current		1140	1400	mA	A, G
I(ADDPLL)	PLL_AVDD current		5	7	mA	H

A Use for budgetary purposes only.

B Assumes typical tile and I/O voltages with no switching activity.

C Includes PLL current.

D Assumes typical tile and I/O voltages with nominal switching activity.

E Assumes 1 MHz = 1 MIPS.

F PD(TYP) value is the usage power consumption under typical operating conditions.

G Measurement conditions: VDD = 1.0 V, VDDIO = 3.3 V, 25 °C, 500 MHz, average device resource usage.

H PLL_AVDD = 1.0 V

The tile power consumption of the device is highly application dependent and should be used for budgetary purposes only.

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.

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.

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.

0x12: Debug SSP

Bits	Perm	Init	Description
31:0	DRW		Value.

B.15 DGETREG operand 1: 0x13

The resource ID of the logical core whose state is to be read.

0x13: DGETREG operand 1

Bits	Perm	Init	Description
31:8	RO	-	Reserved
7:0	DRW		Thread number to be read

B.16 DGETREG operand 2: 0x14

Register number to be read by DGETREG

0x14: DGETREG operand 2

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4:0	DRW		Register number to be read

B.17 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

0x15: Debug interrupt type

Bits	Perm	Init	Description
31:18	RO	-	Reserved
17:16	DRW		Number of the hardware breakpoint/watchpoint which caused the interrupt (always 0 for =HOST= and =DCALL=). If multiple breakpoints/watchpoints trigger at once, the lowest number is taken.
15:8	DRW		Number of thread which caused the debug interrupt (always 0 in the case of =HOST=).
7:3	RO	-	Reserved
2:0	DRW	0	Indicates the cause of the debug interrupt 1: Host initiated a debug interrupt through JTAG 2: Program executed a DCALL instruction 3: Instruction breakpoint 4: Data watch point 5: Resource watch point

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.

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.

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.

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.

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.

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.

0x1F: Debug source

Bits	Perm	Init	Description
31:5	RO	-	Reserved
4	RW		If set, external pin, is the source of last GlobalDebug event.
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.

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.

0x20 .. 0x28: Link status, direction, 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.
15:12	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.

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

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

E 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 30 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

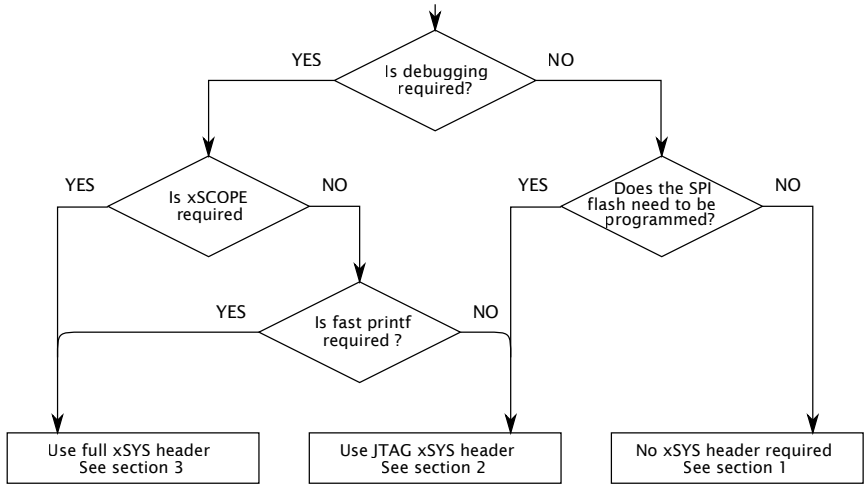

Figure 30: Decision diagram for the xSYS header

E.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.

E.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
- DEBUG_N to pin 11 of the xSYS header

F Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XLF232-1024-FB374. Each of the following sections contains items to check for each design.

F.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 11).

☐ The VDD (core) supply ramps monotonically (rises constantly) from 0V to its final value (0.95V - 1.05V) within 10ms (Section 11).

☐ The VDD (core) supply is capable of supplying 1400 mA (Section 11 and Figure 15).

☐ PLL_AVDD is filtered with a low pass filter, for example an RC filter, see Section 11

F.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 11).

☐ A bulk decoupling capacitor of at least 10uF is placed on each supply (Section 11).

F.3 Power on reset

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

F.4 Clock

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

☐ 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.

F.5 Boot

- ☐ X0D01 has a 1K pull-up to VDDIO (Section 8).
- ☐ The device is kept in reset for at least 1 ms after VDDIO has reached its minimum level (Section 8).

F.6 JTAG, XScope, and debugging

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

F.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 low or not connected (Section 8)
- ☐ Pins X2D04, X2D05, X2D06 and X2D07 are output only and during and after reset, X2D06 is pulled high and X2D04, X2D05, and X2D07 are pulled low (Section 8)

F.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).