Chipsmall
深圳市津利帝科技有限公司
Geelead Technology Limited
Chipsmall
采购部

PRELIMINARY

XEF232-1024-FB374 Datasheet

2015/08/27 Document Number: X009587,

Table of Contents

TO OUR VALUED CUSTOMERS

It is our intention to provide you with accurate and comprehensive documentation for the hardware and software components used in this product. To subscribe to receive updates, visit http://www.xmos.com/.

XMOS Ltd. is the owner or licensee of the information in this document and is providing it to you “AS IS” with no warranty of any kind, express or implied and shall have no liability in relation to its use. XMOS Ltd. makes no representation that the information, or any particular implementation thereof, is or will be free from any claims of infringement and again, shall have no liability in relation to any such claims.

1 xCORE M

32-bit multicore microcon-
inism of the xCORE architec-
nventional microcontrollers,
al-time tasks simultaneously
d network. Because xCORE
ic, you can write software to
ed hardware.

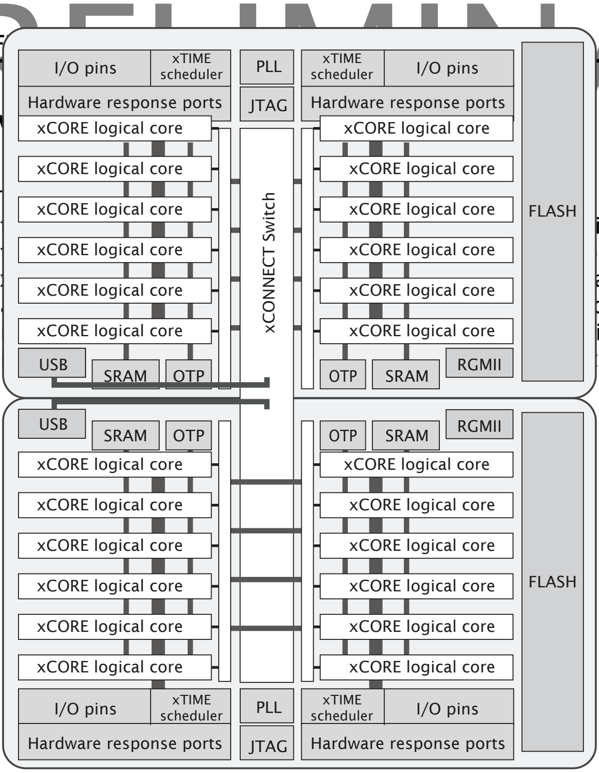


Figure 1: XEF232-1024-FB374 block diagram

Key features of the XEF232-1024-FB374 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

- **Ports** The I/O pins are connected to the processing cores by Hardware Response ports. The port logic can drive its pins high and low, or it can sample the value on its pins optionally waiting for a particular condition. Section 6.3
- **Clock blocks** xCORE devices include a set of programmable clock blocks that can be used to govern the rate at which ports execute. Section 6.4
- **Memory** Each xCORE Tile integrates a bank of SRAM for instructions and data, and a block of one-time programmable (OTP) memory that can be configured for system wide security features. Section 9
- **PLL** The PLL is used to create a high-speed processor clock given a low speed external oscillator. Section 7
- **USB** The USB PHY provides High-Speed and Full-Speed, device, host, and on-the-go functionality. Data is communicated through ports on the digital node. A library is provided to implement USB device functionality. Section 10
- **RGMII** The device has a set of pins that can be dedicated to communicate with an RGMII, including Gbit Ethernet PHYs, according to the RGMII v1.3 specification. Section 11
- **Flash** The device has a built-in 4MBflash. Section 8
- **JTAG** The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 12

1.1 Software

Devices are programmed using C, C++ or xC (C with multicore extensions). XMOS provides tested and proven software libraries, which allow you to quickly add interface and processor functionality such as USB, Ethernet, PWM, graphics driver, and audio EQ to your applications.

1.2 xTIMEcomposer Studio

The xTIMEcomposer Studio development environment provides all the tools you need to write and debug your programs, profile your application, and write images into flash memory or OTP memory on the device. Because xCORE devices operate deterministically, they can be simulated like hardware within xTIMEcomposer: uniquely in the embedded world, xTIMEcomposer Studio therefore includes a static timing analyzer, cycle-accurate simulator, and high-speed in-circuit instrumentation.

xTIMEcomposer can be driven from either a graphical development environment, or the command line. The tools are supported on Windows, Linux and MacOS X and available at no cost from xmos.com/downloads. Information on using the tools is provided in the xTIMEcomposer User Guide, X3766.

2 XEF232-1024-FB374 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - 32 real-time logical cores on 4 xCORE tiles
  - Cores share up to 2000 MIPS
    - — Up to 4000 MIPS in dual issue mode
  - Each logical core has:
    - — Guaranteed throughput of between 1/5 and 1/6 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**
- **RGMII support, compliant with RGMII v1.3 specification**
- **Programmable I/O**
  - 176 general-purpose I/O pins, configurable as input or output
    - — Up to 59 x 1bit port, 22 x 4bit port, 15 x 8bit port, 8 x 16bit port, 4 x 32bit port
    - — 10 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 64 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 1024KB internal single-cycle SRAM (max 256KB per tile) for code and data storage
  - 32KB internal OTP (max 8KB per tile) for application boot code
  - 4MB internal flash for application code and overlays
- **Hardware resources**
  - 24 clock blocks (6 per tile)
  - 40 timers (10 per tile)
  - 16 locks (4 per tile)
- **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**
  - 40: 2000 MIPS
- **Power Consumption**
  - 1140 mA (typical)
- **374-pin FBGA package 0.8 mm pitch**

3 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A	GND	VDDIO	1D X1D11	4E X1D32 rx2	4E X1D26 tx_clk	VDDIOT_0	8D X1D41 tx2	4F X0D31	4F X0D29	TDI	VDDIO	CLK	TDO	4E X3D32 rx2	4F X3D30 rx0	VDDIOT_1	4F X2D31	4F X2D29	4E X2D32	VDDIO	GND
B	1N X0D37 XL_0^{i4}	1M X0D36	1C X1D10	4E X1D33 rx3	4E X1D27 tx_ctl	8D X1D42 tx1	8D X1D40 tx3	4F X0D30	4F X0D28	1M X2D36	GND	RST_N	TCK	4E X3D33 rx3	4F X3D31 rx1	4E X3D27 tx_ctl	4F X2D30	4F X2D28	4E X2D27 XL_7^{o4}	4E X2D26 XL_7^{o3}	1L X2D35 XL_7^{o2}
C	1P X0D39 XL_0^{i2}	1O X0D38 XL_0^{i3}	VDD	4F X1D30 rx0	4F X1D28 rx_clk	8D X1D43 tx0	GND	4E X0D33	4E X0D32	MODE1	OTP_VCC	TRST	1C X3D10	4F X3D29 rx_ctl	GND	8D X3D43 tx0	8D X3D41 tx2	4E X2D33	VDD	1J X2D25 XL_7^{o0}	1K X2D34 XL_7^{o1}
D	8D X0D41 XL_0^{i0}	8D X0D40 XL_0^{i1}	1K X1D34 XL_0^{o2}	4F X1D31 rx1	4F X1D29 rx_ctl	GND	VDDIO	NC	GLOBAL_DEBUG	MODE0		TMS	1D X3D11	4F X3D28 rx_clk	4E X3D26 tx_clk	8D X3D42 tx1	8D X3D40 tx3	32A X2D70 XL_6^{o4}	1A X3D00 XL_7^{i2}	1B X3D01 XL_7^{i1}	1I X2D24 XL_7^{i0}
E	8D X0D43 XL_0^{o1}	8D X0D42 XL_0^{o0}	1L X1D35 XL_0^{o3}	VDD	VDD	GND	VDDIO	VDD	VDD				VDD	VDD	VDDIO	GND	VDD	VDD	32A X2D69 XL_6^{o3}	4A X3D08 XL_7^{i4}	4A X3D09 XL_7^{i3}
F	1M X1D36 XL_0^{o4}	VDDIO	GND	VDD	VDD	VDD	VDD	VDD	VDD	PLL_AGND	PLL_AVDD	VDD	VDD	VDD	VDD	VDD	VDD	VDD	GND	VDDIO	32A X2D68 XL_6^{o2}
G	32A X1D49 XL_1^{i4}	32A X1D50 XL_1^{i3}	32A X1D51 XL_1^{i2}	NC	NC	1O NC XL_0^{i3}	1N NC XL_0^{i4}	1K NC XL_0^{o2}	8D NC XL_0^{o1}				8D NC XL_0^{o0}	8D NC XL_0^{i0}	8D NC XL_0^{i1}	1P NC XL_0^{i2}	NC	NC	32A X2D67 XL_6^{o1}	32A X2D66 XL_6^{o0}	32A X2D65 XL_6^{i0}
H	32A X1D53 XL_1^{i0}	32A X1D52 XL_1^{i1}	VDD																VDD	32A X2D63 XL_6^{i2}	32A X2D64 XL_6^{i1}
J	32A X1D54 XL_1^{o0}	32A X1D55 XL_1^{o1}	VDD		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		VDD	32A X2D62 XL_6^{i3}	32A X2D61 XL_6^{i4}
K	32A X1D58 XL_1^{o4}	32A X1D57 XL_1^{o3}	32A X1D56 XL_1^{o2}		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		32A X2D56 XL_5^{o2}	32A X2D57 XL_5^{o3}	32A X2D58 XL_5^{o4}
L	VDDIO	GND	32A X1D61 XL_2^{i4}		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		32A X2D55 XL_5^{o1}	GND	VDDIO
M	32A X1D64 XL_2^{i1}	32A X1D63 XL_2^{i2}	32A X1D62 XL_2^{i3}		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		32A X2D54 XL_5^{o0}	32A X2D53 XL_5^{i0}	32A X2D52 XL_5^{i1}
N	32A X1D65 XL_2^{i0}	32A X1D66 XL_2^{o0}	VDD		GND	GND	GND	GND	GND				GND	GND	GND	GND	GND		VDD	32A X2D50 XL_5^{i3}	32A X2D51 XL_5^{i2}
P	32A X1D68 XL_2^{o2}	32A X1D67 XL_2^{o1}	VDD																VDD	4B X3D06 XL_4^{o3}	4B X3D07 XL_4^{o4}
R	32A X1D69 XL_2^{o3}	32A X1D70 XL_2^{o4}	1N X1D37 XL_3^{i4}	NC	NC	NC	NC	4D NC XL_3^{o1}	1B NC XL_3^{o2}				1G NC XL_3^{o4}	1C NC XL_3^{o3}	NC	NC	NC	NC	32A X2D49 XL_5^{i4}	4B X3D04 XL_4^{o1}	4B X3D05 XL_4^{o2}
T	1O X1D38 XL_3^{i3}	VDDIO	GND	VDD	VDD	VDD	USB_VDD_0	VDD	VDD	VDD	GND	VDD	VDD	VDD	USB_VDD_1	VDD	VDD	VDD	GND	VDDIO	4A X3D03 XL_4^{o0}
U	4D X1D17 XL_3^{i0}	4D X1D16 XL_3^{i1}	1P X1D39 XL_3^{i2}	VDD	VDD	GND	VDDIO	NC	VDD		VDDIO		VDD	VDD	VDDIO	GND	VDD	VDD	NC	4D X2D19 XL_4^{i1}	4A X3D02 XL_4^{i0}
V	4D X1D19 XL_3^{o1}	4D X1D18 XL_3^{o0}	1B X0D01 XL_3^{o2}	4A X0D02	4A X0D08	1D X0D11	USB_ID_0	4C X1D14	1J X1D25	4C X0D21	NC	1H X3D23	4B X2D05	4B X2D07	USB_ID_1	NC	4C X3D15	4C X3D21	1E X2D12	4D X2D17 XL_4^{i3}	4D X2D18 XL_4^{i2}
W	1C X0D10 XL_3^{o3}	1G X1D22 XL_3^{o4}	USB_VDD33_0	4A X0D03	4A X0D09	USB_RTUNE_0	GND	4C X1D15	4C X0D14	1E X0D12	1H X0D23	1A X2D00	4B X2D04	4B X2D06	GND	USB_RTUNE_1	4C X3D14	4C X3D20	USB_VDD33_1	1H X2D23	4D X2D16 XL_4^{i4}
Y	1H X1D23	1A X0D00	4B X0D04	4B X0D06	1E X1D12	USB_VBUS_0	1I X1D24	4C X1D20	4C X0D15	1F X0D13	GND	1D X2D11	4A X2D02	4A X2D08	1F X3D13	VUSB_BUS_1	4C X2D14	4C X2D20	1I X3D24	1F X2D13	1G X2D22
AA	GND	VDDIO	4B X0D05	4B X0D07	1F X1D13	USB_DM_0	USB_DP_0	4C X1D21	4C X0D20	1G X0D22	VDDIO	1E X3D12	4A X2D03	4A X2D09	USB_DM_1	USB_DP_1	4C X2D15	4C X2D21	1J X3D25	VDDIO	GND

4 Signal Description

This section lists the signals and I/O pins available on the XEF232-1024-FB374. The device provides a combination of 1bit, 4bit, 8bit and 16bit ports, as well as wider ports that are fully or partially (gray) bonded out. All pins of a port provide either output or input, but signals in different directions cannot be mapped onto the same port.

Pins may have one or more of the following properties:

- PD/PU: The IO pin a weak pull-down or pull-up resistor. On GPIO pins this resistor can be enabled.
- ST: The IO pin has a Schmitt Trigger on its input.

Power pins (12)

Signal	Function	Type	Properties
GND	Digital ground	GND	
OTP_VCC	OTP power supply	PWR	
PLL_AGND	Analog ground for PLL	PWR	
PLL_AVDD	Analog PLL power	PWR	
USB_VDD33_0		PWR	
USB_VDD33_1		PWR	
USB_VDD_0		PWR	
USB_VDD_1		PWR	
VDD	Digital tile power	PWR	
VDDIO	Digital I/O power	PWR	
VDDIOT_0		PWR	
VDDIOT_1		PWR	

JTAG pins (6)

Signal	Function	Type	Properties
RST_N	Global reset input	Input	IOL, PU, ST
TCK	Test clock	Input	IOL, PD, ST
TDI	Test data input	Input	IOL, PU
TDO	Test data output	Output	IOL, PD
TMS	Test mode select	Input	IOL, PU
TRST		Input	IOL, PU, ST

I/O pins (176)

Signal	Function		Type	Properties
X0D00		$1A^0$	I/O	IOL, PD
X0D01	$XL3^2_{out}$	$1B^0$	I/O—	IOL, PD

(continued)

Signal	Function						Type	Properties
X0D02			$4A^{0}$	$8A^{0}$	$16A^{0}$	$32A^{20}$	I/O	IOL, PD
X0D03			$4A^{1}$	$8A^{1}$	$16A^{1}$	$32A^{21}$	I/O	IOL, PD
X0D04			$4B^{0}$	$8A^{2}$	$16A^{2}$	$32A^{22}$	I/O—	IOL, PD
X0D05			$4B^{1}$	$8A^{3}$	$16A^{3}$	$32A^{23}$	I/O—	IOL, PD
X0D06			$4B^{2}$	$8A^{4}$	$16A^{4}$	$32A^{24}$	I/O—	IOL, PD
X0D07			$4B^{3}$	$8A^{5}$	$16A^{5}$	$32A^{25}$	I/O—	IOL, PD
X0D08			$4A^{2}$	$8A^{6}$	$16A^{6}$	$32A^{26}$	I/O	IOL, PD
X0D09			$4A^{3}$	$8A^{7}$	$16A^{7}$	$32A^{27}$	I/O	IOL, PD
X0D10	$XL3^{3}_{out}$	$1C^{0}$					I/O—	IOL, PD
X0D11		$1D^{0}$					I/O	IOL, PD
X0D12		$1E^{0}$					I/O	IOR, PD
X0D13		$1F^{0}$					I/O	IOR, PD
X0D14			$4C^{0}$	$8B^{0}$	$16A^{8}$	$32A^{28}$	I/O	IOR, PD
X0D15			$4C^{1}$	$8B^{1}$	$16A^{9}$	$32A^{29}$	I/O	IOR, PD
X0D20			$4C^{2}$	$8B^{6}$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X0D21			$4C^{3}$	$8B^{7}$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X0D22		$1G^{0}$					I/O	IOR, PD
X0D23		$1H^{0}$					I/O	IOR, PD
X0D28			$4F^{0}$	$8C^{2}$	$16B^{2}$		I/O	IOR, PD
X0D29			$4F^{1}$	$8C^{3}$	$16B^{3}$		I/O	IOR, PD
X0D30			$4F^{2}$	$8C^{4}$	$16B^{4}$		I/O	IOR, PD
X0D31			$4F^{3}$	$8C^{5}$	$16B^{5}$		I/O	IOR, PD
X0D32			$4E^{2}$	$8C^{6}$	$16B^{6}$		I/O	IOR, PD
X0D33			$4E^{3}$	$8C^{7}$	$16B^{7}$		I/O	IOR, PD
X0D36		$1M^{0}$		$8D^{0}$	$16B^{8}$		I/O	IOL, PD
X0D37	$XL0^{4}_{in}$	$1N^{0}$		$8D^{1}$	$16B^{9}$		I/O	IOL, PD
X0D38	$XL0^{3}_{in}$	$1O^{0}$		$8D^{2}$	$16B^{10}$		I/O	IOL, PD
X0D39	$XL0^{2}_{in}$	$1P^{0}$		$8D^{3}$	$16B^{11}$		I/O	IOL, PD
X0D40	$XL0^{1}_{in}$			$8D^{4}$	$16B^{12}$		I/O	IOL, PD
X0D41	$XL0^{0}_{in}$			$8D^{5}$	$16B^{13}$		I/O	IOL, PD
X0D42	$XL0^{0}_{out}$			$8D^{6}$	$16B^{14}$		I/O	IOL, PD
X0D43	$XL0^{1}_{out}$			$8D^{7}$	$16B^{15}$		I/O	IOL, PD
X1D10		$1C^{0}$					I/O	IOT, PD
X1D11		$1D^{0}$					I/O	IOT, PD
X1D12		$1E^{0}$					I/O	IOL, PD
X1D13		$1F^{0}$					I/O	IOL, PD
X1D14			$4C^{0}$	$8B^{0}$	$16A^{8}$	$32A^{28}$	I/O	IOR, PD
X1D15			$4C^{1}$	$8B^{1}$	$16A^{9}$	$32A^{29}$	I/O	IOR, PD
X1D16	$XL3^{1}_{in}$		$4D^{0}$	$8B^{2}$	$16A^{10}$		I/O	IOL, PD
X1D17	$XL3^{0}_{in}$		$4D^{1}$	$8B^{3}$	$16A^{11}$		I/O	IOL, PD
X1D18	$XL3^{0}_{out}$		$4D^{2}$	$8B^{4}$	$16A^{12}$		I/O	IOL, PD
X1D19	$XL3^{1}_{out}$		$4D^{3}$	$8B^{5}$	$16A^{13}$		I/O	IOL, PD
X1D20			$4C^{2}$	$8B^{6}$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD

(continued)

Signal	Function						Type	Properties
X1D21			4C^{3}	8B^{7}	16A^{15}	32A^{31}	I/O	IOR, PD
X1D22	XL3$^{4}_{out}$	1G^{0}					I/O	IOL, PD
X1D23		1H^{0}					I/O	IOL, PD
X1D24		1I^{0}					I/O	IOR, PD
X1D25		1J^{0}					I/O	IOR, PD
X1D26	tx_clk (rgmii)		4E^{0}	8C^{0}	16B^{0}		I/O	IOT, PD
X1D27	tx_ctl (rgmii)		4E^{1}	8C^{1}	16B^{1}		I/O	IOT, PD
X1D28	rx_clk (rgmii)		4F^{0}	8C^{2}	16B^{2}		I/O	IOT, PD
X1D29	rx_ctl (rgmii)		4F^{1}	8C^{3}	16B^{3}		I/O	IOT, PD
X1D30	rx0 (rgmii)		4F^{2}	8C^{4}	16B^{4}		I/O	IOT, PD
X1D31	rx1 (rgmii)		4F^{3}	8C^{5}	16B^{5}		I/O	IOT, PD
X1D32	rx2 (rgmii)		4E^{2}	8C^{6}	16B^{6}		I/O	IOT, PD
X1D33	rx3 (rgmii)		4E^{3}	8C^{7}	16B^{7}		I/O	IOT, PD
X1D34	XL0$^{2}_{out}$	1K^{0}					I/O	IOL, PD
X1D35	XL0$^{3}_{out}$	1L^{0}					I/O	IOL, PD
X1D36	XL0$^{4}_{out}$	1M^{0}		8D^{0}	16B^{8}		I/O	IOL, PD
X1D37	XL3$^{4}_{in}$	1N^{0}		8D^{1}	16B^{9}		I/O	IOL, PD
X1D38	XL3$^{3}_{in}$	1O^{0}		8D^{2}	16B^{10}		I/O	IOL, PD
X1D39	XL3$^{2}_{in}$	1P^{0}		8D^{3}	16B^{11}		I/O	IOL, PD
X1D40	tx3 (rgmii)			8D^{4}	16B^{12}		I/O	IOT, PD
X1D41	tx2 (rgmii)			8D^{5}	16B^{13}		I/O	IOT, PD
X1D42	tx1 (rgmii)			8D^{6}	16B^{14}		I/O	IOT, PD
X1D43	tx0 (rgmii)			8D^{7}	16B^{15}		I/O	IOT, PD
X1D49	XL1$^{4}_{in}$					32A^{0}	I/O	IOL, PD
X1D50	XL1$^{3}_{in}$					32A^{1}	I/O	IOL, PD
X1D51	XL1$^{2}_{in}$					32A^{2}	I/O	IOL, PD
X1D52	XL1$^{1}_{in}$					32A^{3}	I/O	IOL, PD
X1D53	XL1$^{0}_{in}$					32A^{4}	I/O	IOL, PD
X1D54	XL1$^{0}_{out}$					32A^{5}	I/O	IOL, PD
X1D55	XL1$^{1}_{out}$					32A^{6}	I/O	IOL, PD
X1D56	XL1$^{2}_{out}$					32A^{7}	I/O	IOL, PD
X1D57	XL1$^{3}_{out}$					32A^{8}	I/O	IOL, PD
X1D58	XL1$^{4}_{out}$					32A^{9}	I/O	IOL, PD
X1D61	XL2$^{4}_{in}$					32A^{10}	I/O	IOL, PD
X1D62	XL2$^{3}_{in}$					32A^{11}	I/O	IOL, PD
X1D63	XL2$^{2}_{in}$					32A^{12}	I/O	IOL, PD
X1D64	XL2$^{1}_{in}$					32A^{13}	I/O	IOL, PD
X1D65	XL2$^{0}_{in}$					32A^{14}	I/O	IOL, PD
X1D66	XL2$^{0}_{out}$					32A^{15}	I/O	IOL, PD
X1D67	XL2$^{1}_{out}$					32A^{16}	I/O	IOL, PD
X1D68	XL2$^{2}_{out}$					32A^{17}	I/O	IOL, PD
X1D69	XL2$^{3}_{out}$					32A^{18}	I/O	IOL, PD
X1D70	XL2$^{4}_{out}$					32A^{19}	I/O	IOL, PD

(continued)

Signal	Function						Type	Properties
X2D00		$1A^0$					I/O	IOL, PD
X2D02			$4A^0$	$8A^0$	$16A^0$	$32A^{20}$	I/O	IOL, PD
X2D03			$4A^1$	$8A^1$	$16A^1$	$32A^{21}$	I/O	IOL, PD
X2D04			$4B^0$	$8A^2$	$16A^2$	$32A^{22}$	I/O	IOL, PD
X2D05			$4B^1$	$8A^3$	$16A^3$	$32A^{23}$	I/O	IOL, PD
X2D06			$4B^2$	$8A^4$	$16A^4$	$32A^{24}$	I/O	IOL, PD
X2D07			$4B^3$	$8A^5$	$16A^5$	$32A^{25}$	I/O	IOL, PD
X2D08			$4A^2$	$8A^6$	$16A^6$	$32A^{26}$	I/O	IOL, PD
X2D09			$4A^3$	$8A^7$	$16A^7$	$32A^{27}$	I/O	IOL, PD
X2D11		$1D^0$					I/O	IOL, PD
X2D12		$1E^0$					I/O	IOR, PD
X2D13		$1F^0$					I/O	IOR, PD
X2D14			$4C^0$	$8B^0$	$16A^8$	$32A^{28}$	I/O	IOR, PD
X2D15			$4C^1$	$8B^1$	$16A^9$	$32A^{29}$	I/O	IOR, PD
X2D16	$XL4^4_{in}$		$4D^0$	$8B^2$	$16A^{10}$		I/O	IOR, PD
X2D17	$XL4^3_{in}$		$4D^1$	$8B^3$	$16A^{11}$		I/O	IOR, PD
X2D18	$XL4^2_{in}$		$4D^2$	$8B^4$	$16A^{12}$		I/O	IOR, PD
X2D19	$XL4^1_{in}$		$4D^3$	$8B^5$	$16A^{13}$		I/O	IOR, PD
X2D20			$4C^2$	$8B^6$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X2D21			$4C^3$	$8B^7$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X2D22		$1G^0$					I/O	IOR, PD
X2D23		$1H^0$					I/O	IOR, PD
X2D24	$XL7^0_{in}$	$1I^0$					I/O	IOR, PD
X2D25	$XL7^0_{out}$	$1J^0$					I/O	IOR, PD
X2D26	$XL7^3_{out}$		$4E^0$	$8C^0$	$16B^0$		I/O	IOR, PD
X2D27	$XL7^4_{out}$		$4E^1$	$8C^1$	$16B^1$		I/O	IOR, PD
X2D28			$4F^0$	$8C^2$	$16B^2$		I/O	IOR, PD
X2D29			$4F^1$	$8C^3$	$16B^3$		I/O	IOR, PD
X2D30			$4F^2$	$8C^4$	$16B^4$		I/O	IOR, PD
X2D31			$4F^3$	$8C^5$	$16B^5$		I/O	IOR, PD
X2D32			$4E^2$	$8C^6$	$16B^6$		I/O	IOR, PD
X2D33			$4E^3$	$8C^7$	$16B^7$		I/O	IOR, PD
X2D34	$XL7^1_{out}$	$1K^0$					I/O	IOR, PD
X2D35	$XL7^2_{out}$	$1L^0$					I/O	IOR, PD
X2D36		$1M^0$		$8D^0$	$16B^8$		I/O	IOL, PD
X2D49	$XL5^4_{in}$					$32A^0$	I/O	IOR, PD
X2D50	$XL5^3_{in}$					$32A^1$	I/O	IOR, PD
X2D51	$XL5^2_{in}$					$32A^2$	I/O	IOR, PD
X2D52	$XL5^1_{in}$					$32A^3$	I/O	IOR, PD
X2D53	$XL5^0_{in}$					$32A^4$	I/O	IOR, PD
X2D54	$XL5^0_{out}$					$32A^5$	I/O	IOR, PD
X2D55	$XL5^1_{out}$					$32A^6$	I/O	IOR, PD
X2D56	$XL5^2_{out}$					$32A^7$	I/O	IOR, PD

(continued)

Signal	Function						Type	Properties
X2D57	$XL5^{3}_{out}$					$32A^{8}$	I/O	IOR, PD
X2D58	$XL5^{4}_{out}$					$32A^{9}$	I/O	IOR, PD
X2D61	$XL6^{4}_{in}$					$32A^{10}$	I/O	IOR, PD
X2D62	$XL6^{3}_{in}$					$32A^{11}$	I/O	IOR, PD
X2D63	$XL6^{2}_{in}$					$32A^{12}$	I/O	IOR, PD
X2D64	$XL6^{1}_{in}$					$32A^{13}$	I/O	IOR, PD
X2D65	$XL6^{0}_{in}$					$32A^{14}$	I/O	IOR, PD
X2D66	$XL6^{0}_{out}$					$32A^{15}$	I/O	IOR, PD
X2D67	$XL6^{1}_{out}$					$32A^{16}$	I/O	IOR, PD
X2D68	$XL6^{2}_{out}$					$32A^{17}$	I/O	IOR, PD
X2D69	$XL6^{3}_{out}$					$32A^{18}$	I/O	IOR, PD
X2D70	$XL6^{4}_{out}$					$32A^{19}$	I/O	IOR, PD
X3D00	$XL7^{2}_{in}$	$1A^{0}$					I/O	IOR, PD
X3D01	$XL7^{1}_{in}$	$1B^{0}$					I/O	IOR, PD
X3D02	$XL4^{0}_{in}$		$4A^{0}$	$8A^{0}$	$16A^{0}$	$32A^{20}$	I/O	IOR, PD
X3D03	$XL4^{0}_{out}$		$4A^{1}$	$8A^{1}$	$16A^{1}$	$32A^{21}$	I/O	IOR, PD
X3D04	$XL4^{1}_{out}$		$4B^{0}$	$8A^{2}$	$16A^{2}$	$32A^{22}$	I/O	IOR, PD
X3D05	$XL4^{2}_{out}$		$4B^{1}$	$8A^{3}$	$16A^{3}$	$32A^{23}$	I/O	IOR, PD
X3D06	$XL4^{3}_{out}$		$4B^{2}$	$8A^{4}$	$16A^{4}$	$32A^{24}$	I/O	IOR, PD
X3D07	$XL4^{4}_{out}$		$4B^{3}$	$8A^{5}$	$16A^{5}$	$32A^{25}$	I/O	IOR, PD
X3D08	$XL7^{4}_{in}$		$4A^{2}$	$8A^{6}$	$16A^{6}$	$32A^{26}$	I/O	IOR, PD
X3D09	$XL7^{3}_{in}$		$4A^{3}$	$8A^{7}$	$16A^{7}$	$32A^{27}$	I/O	IOR, PD
X3D10		$1C^{0}$					I/O	IOT, PD
X3D11		$1D^{0}$					I/O	IOT, PD
X3D12		$1E^{0}$					I/O	IOL, PD
X3D13		$1F^{0}$					I/O	IOL, PD
X3D14			$4C^{0}$	$8B^{0}$	$16A^{8}$	$32A^{28}$	I/O	IOR, PD
X3D15			$4C^{1}$	$8B^{1}$	$16A^{9}$	$32A^{29}$	I/O	IOR, PD
X3D20			$4C^{2}$	$8B^{6}$	$16A^{14}$	$32A^{30}$	I/O	IOR, PD
X3D21			$4C^{3}$	$8B^{7}$	$16A^{15}$	$32A^{31}$	I/O	IOR, PD
X3D23		$1H^{0}$					I/O	IOL, PD
X3D24		$1I^{0}$					I/O	IOR, PD
X3D25		$1J^{0}$					I/O	IOR, PD
X3D26	tx_clk (rgmii)		$4E^{0}$	$8C^{0}$	$16B^{0}$		I/O	IOT, PD
X3D27	tx_ctl (rgmii)		$4E^{1}$	$8C^{1}$	$16B^{1}$		I/O	IOT, PD
X3D28	rx_clk (rgmii)		$4F^{0}$	$8C^{2}$	$16B^{2}$		I/O	IOT, PD
X3D29	rx_ctl (rgmii)		$4F^{1}$	$8C^{3}$	$16B^{3}$		I/O	IOT, PD
X3D30	rx0 (rgmii)		$4F^{2}$	$8C^{4}$	$16B^{4}$		I/O	IOT, PD
X3D31	rx1 (rgmii)		$4F^{3}$	$8C^{5}$	$16B^{5}$		I/O	IOT, PD
X3D32	rx2 (rgmii)		$4E^{2}$	$8C^{6}$	$16B^{6}$		I/O	IOT, PD
X3D33	rx3 (rgmii)		$4E^{3}$	$8C^{7}$	$16B^{7}$		I/O	IOT, PD
X3D40	tx3 (rgmii)			$8D^{4}$	$16B^{12}$		I/O	IOT, PD
X3D41	tx2 (rgmii)			$8D^{5}$	$16B^{13}$		I/O	IOT, PD

(continued)

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

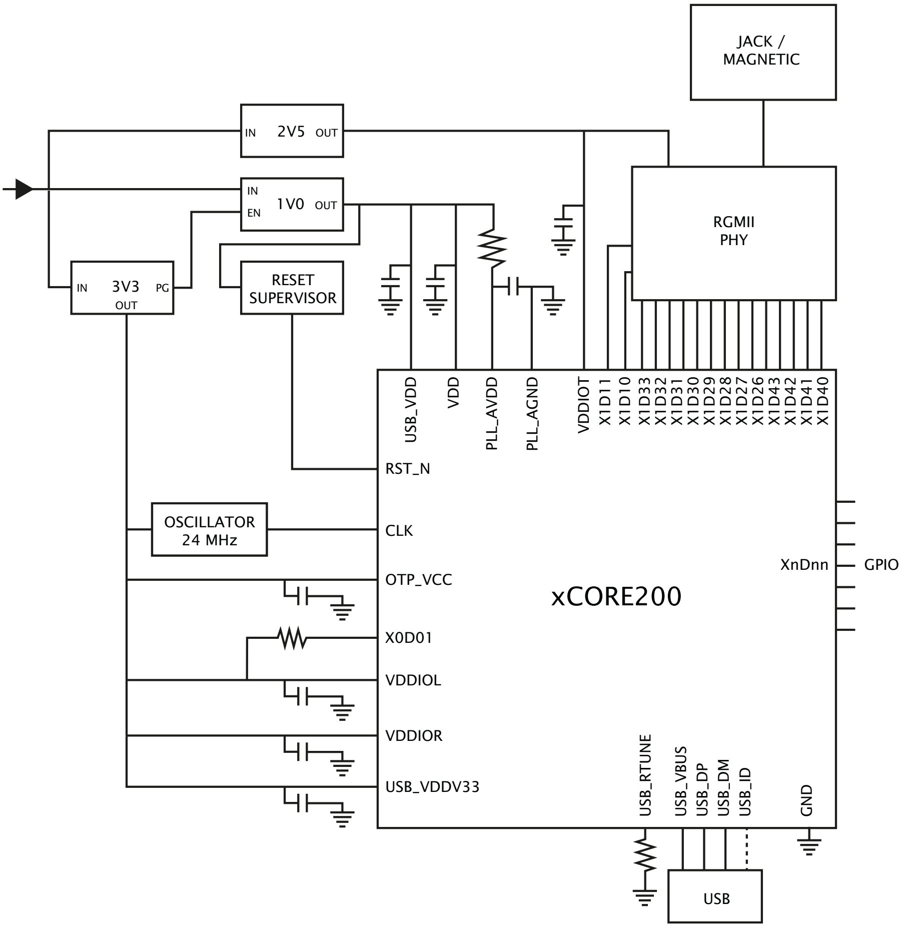


Figure 2: Simplified Reference Schematic

6 Product Overview

The XEF232-1024-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 6 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 XEF232-1024-FB374, and the software running on it. A combination of 1bit, 4bit, 8bit, 16bit and 32bit ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.

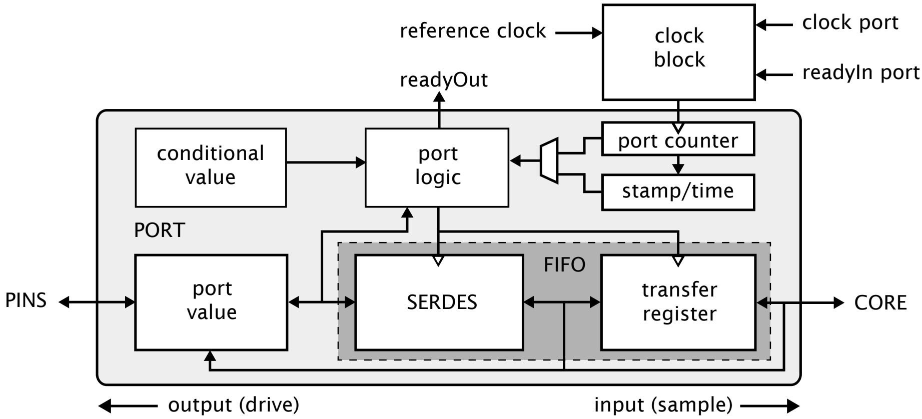


Figure 4: Port block diagram

The port logic can drive its pins high or low, or it can sample the value on its pins, optionally waiting for a particular condition. Ports are accessed using dedicated instructions that are executed in a single processor cycle. xCORE-200 IO pins can be used as *open collector* outputs, where signals are driven low if a zero is output, but left high impedance if a one is output. This option is set on a per-port basis.

Data is transferred between the pins and core using a FIFO that comprises a SERDES and transfer register, providing options for serialization and buffered data.

Each port has a 16-bit counter that can be used to control the time at which data is transferred between the port value and transfer register. The counter values can be obtained at any time to find out when data was obtained, or used to delay I/O until some time in the future. The port counter value is automatically saved as a timestamp, that can be used to provide precise control of response times.

The ports and xCONNECT links are multiplexed onto the physical pins. If an xConnect Link is enabled, the pins of the underlying ports are disabled. If a port is enabled, it overrules ports with higher widths that share the same pins. The pins on the wider port that are not shared remain available for use when the narrower port is enabled. Ports always operate at their specified width, even if they share pins with another port.

6.4 Clock blocks

xCORE devices include a set of programmable clocks called clock blocks that can be used to govern the rate at which ports execute. Each xCORE tile has six clock blocks: the first clock block provides the tile reference clock and runs at a default frequency of 100MHz; the remaining clock blocks can be set to run at different frequencies.

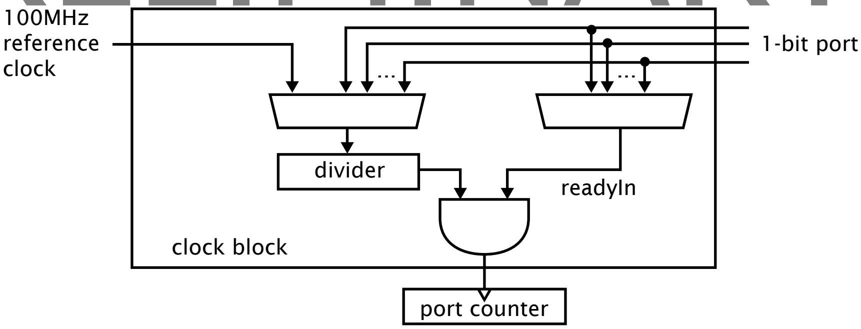


Figure 5: Clock block diagram

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

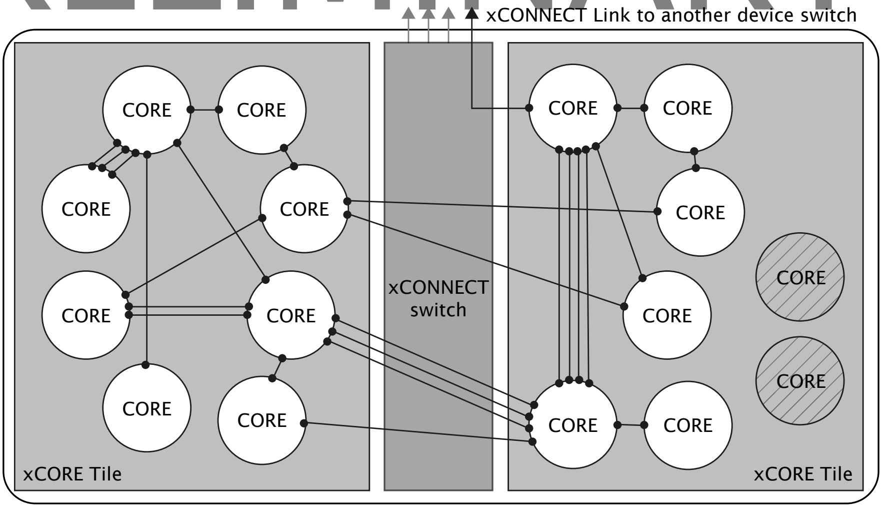


Figure 6: Switch, links and channel ends

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-UEF 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 PLL multiplication value is selected through the two MODE pins, and can be changed by software to speed up the tile or use less power. The MODE pins are set as shown in Figure 7:

Figure 7: PLL multiplier values and MODE pins

Oscillator	MODE		Tile	PLL Ratio	PLL settings		
Frequency	1	0	Frequency		*OD*	*F*	*R*
3.25-10 MHz	0	0	130-400 MHz	40	1	159	0
9-25 MHz	1	1	144-400 MHz	16	1	63	0
25-50 MHz	1	0	167-400 MHz	8	1	31	0
50-100 MHz	0	1	196-400 MHz	4	1	15	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.

The MODE pins must be held at a static value during and after deassertion of the system reset. If the USB PHY is used, then either a 24 MHz or 12 MHz oscillator must be used.

If a different tile frequency is required (eg, 500 MHz), then the PLL must be reprogrammed after boot to provide the required tile frequency. The XMOS tools perform this operation by default. Further details on configuring the clock can be found in the xCORE-200 Clock Frequency Control document.

8 Boot Procedure

The device is kept in reset by driving RST_N low. When in reset, all GPIO pins have a pull-down enabled. The processor must be held in reset until VDDIOL is in spec for at least 1 ms. When the device is taken out of reset by releasing RST_N the processor starts its internal reset process. After 15-150 μs (depending on the input clock) the processor boots.

The device boots from a QSPI flash that is embedded in the device. The QSPI flash is connected to the ports on Tile 0 as shown in Figure 8. An external 1K resistor must connect X0D01 to VDDIOL. X0D10 should ideally not be connected. If X0D10 is connected, then a 150 ohm series resistor close to the device is recommended. X0D04..X0D07 should be not connected.

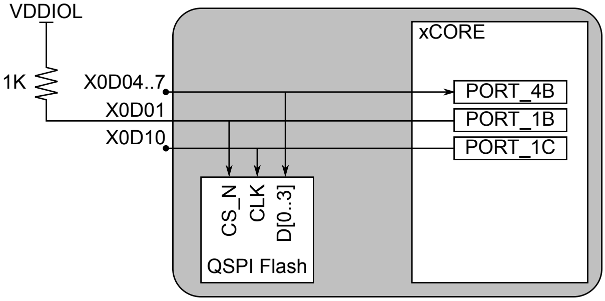


Figure 8: QSPI port connectivity

The xCORE Tile boot procedure is illustrated in Figure 9. If bit 5 of the security register (*see* §9.1) is set, the device boots from OTP. Otherwise, the device boots from the internal flash.

The boot image has the following format:

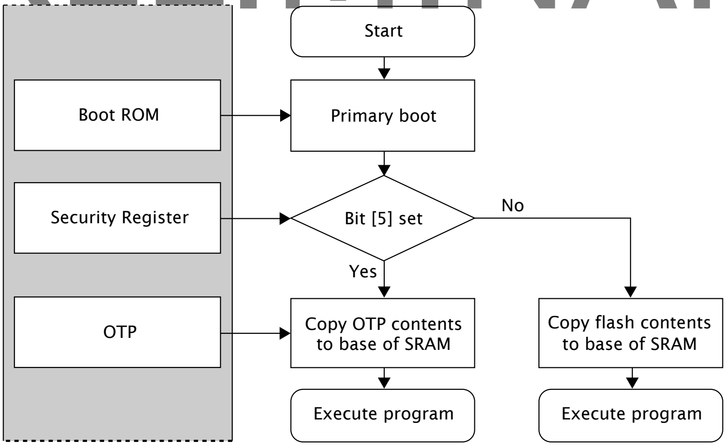


Figure 9: Boot procedure

- A 32-bit program size s in words.
- Program consisting of $s \times 4$ bytes.
- A 32-bit CRC, or the value 0x0D15AB1E to indicate that no CRC check should be performed.

The program size and CRC are stored least significant byte first. The program is loaded into the lowest memory address of RAM, and the program is started from that address. The CRC is calculated over the byte stream represented by the program size and the program itself. The polynomial used is 0xEDB88320 (IEEE 802.3); the CRC register is initialized with 0xFFFFFFFF and the residue is inverted to produce the CRC.

8.1 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 10 provide a strong level of protection and are sufficient for providing strong IP security.

9 Memory

9.1 OTP

Each 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.

Feature	Bit	Description
Disable JTAG	0	The JTAG interface is disabled, making it impossible for the tile state or memory content to be accessed via the JTAG interface.
Disable Link access	1	Other tiles are forbidden access to the processor state via the system switch. Disabling both JTAG and Link access transforms an xCORE Tile into a “secure island” with other tiles free for non-secure user application code.
Secure Boot	5	The xCORE Tile is forced to boot from address 0 of the OTP, allowing the xCORE Tile boot ROM to be bypassed (*see* §8).
Redundant rows	7	Enables redundant rows in OTP.
Sector Lock 0	8	Disable programming of OTP sector 0.
Sector Lock 1	9	Disable programming of OTP sector 1.
Sector Lock 2	10	Disable programming of OTP sector 2.
Sector Lock 3	11	Disable programming of OTP sector 3.
OTP Master Lock	12	Disable OTP programming completely: disables updates to all sectors and security register.
Disable JTAG-OTP	13	Disable all (read & write) access from the JTAG interface to this OTP.
	21..15	General purpose software accessable security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 10: Security register features

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

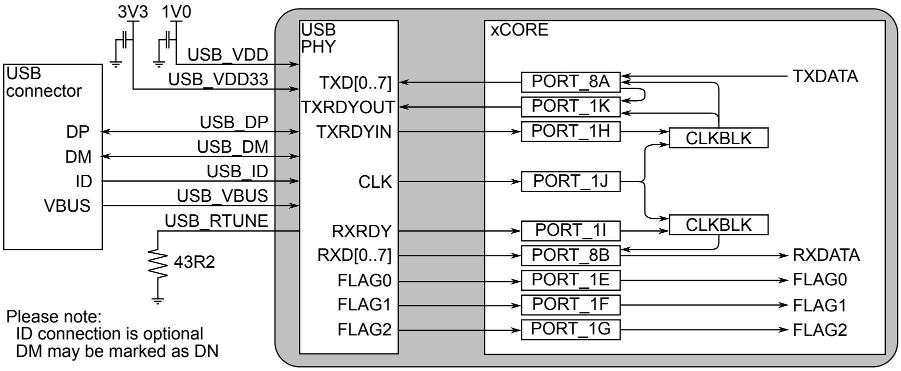
Each xCORE Tile integrates a single 256KB SRAM bank for both instructions and data. All internal memory is 32 bits wide, and instructions are either 16-bit or 32-bit. Byte (8-bit), half-word (16-bit) or word (32-bit) accesses are supported and are executed within one tile clock cycle. There is no dedicated 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, libxud_s.a, is provided to implement USB device functionality.

The USB PHY is connected to the ports on Tile 0 and Tile 1 as shown in Figure 11. When the USB PHY is enabled on Tile 0, the ports shown can on Tile 0 only be used with the USB PHY. When the USB PHY is enabled on Tile 1, then the ports shown can on Tile 1 only be used with the USB PHY. All other IO pins and ports are unaffected. The USB PHY should not be enabled on both tiles.

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



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

Figure 11: USB port functions

Figure 11 shows how two clock blocks can be used to clock the USB ports. One clock block for the TXDATA path, and one clock block for the RXDATA path. Details on how to connect those ports are documented in an application note on USB for xCORE-200.

10.1 Logical Core Requirements

The XMOS XUD software component runs in a single logical core with endpoint and application cores communicating with it via a combination of channel communication and shared memory variables.

Each IN (host requests data from device) or OUT (data transferred from host to device) endpoint requires one logical core.

11 RGMII

The device has a series of pins that are dedicated to communicate with an RGMII PHY, as per the RGMII v1.3 spec. This can be used to communicate with GBit Ethernet PHYs. The pins and functions are listed in Figure 12. When RGMII mode is enabled (using processor status register 2) these pins can no longer be used as GPIO pins, and will instead be driven directly from an RGMII block that provides DDR to SDR conversion, which in turn is interfaced to a set of ports on Tile 1.

Pin	RGMII Function	
X1D40	TX3	Transmit bit 3
X1D41	TX2	Transmit bit 2
X1D42	TX1	Transmit bit 1
X1D43	TX0	Transmit bit 0
X1D26	TX_CLK	Receive clock (125 MHz)
X1D27	TX_CTL	Transmit data valid/error
X1D28	RX_CLK	Receive clock (125 MHz)
X1D29	RX_CTL	Receive data valid/error
X1D30	RX0	Receive bit 0
X1D31	RX1	Receive bit 1
X1D32	RX2	Receive bit 2
X1D33	RX3	Receive bit 3

Figure 12: RGMII block pin functions

The RGMII block is connected to the ports on Tile 1 as shown in Figure 13. When the RGMII block is enabled, the ports shown can only be used with the RGMII block, and IO pins X1D26..X1D33/X1D40..X1D43 can only be used with the RGMII block. Other IO pins and ports are unaffected.

The RGMII block generates a clock (configured using processor status register 2), and has the facility to delay the outgoing clock edge, putting it out of phase with the data. The RGMII block translates the double data-rate 4-wire data signals and 1-wire control signal into single-data rate 8-wire TX and DX signals and two control signals. Figure 13 shows how four clock blocks can be used to clock the RGMII ports. One clock block for the TXDATA path, one clock block for the RXDATA path, one clock block to delay the TX_CLK, and one clock block clocked on a negative valid signal to enable mode switching between 10/100/1000 speeds. Details on how to connect those ports are documented in an application note on RGMII for xCORE-200. The XMOS RGMII software component runs a MAC layer on tile 1.

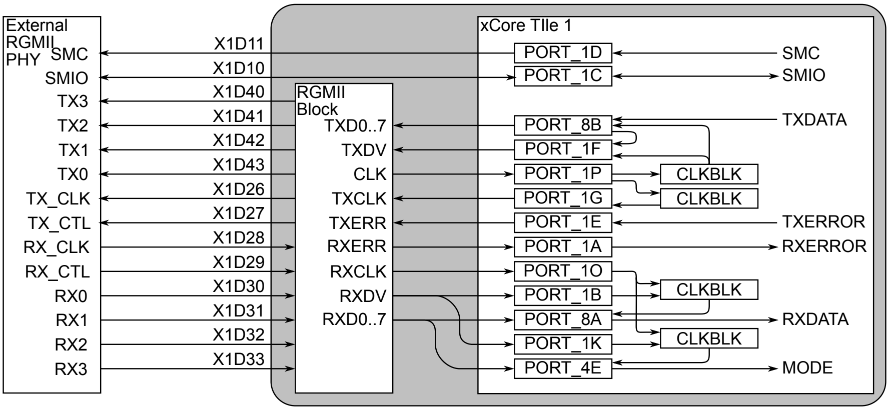


Figure 13: RGMII port functions

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.

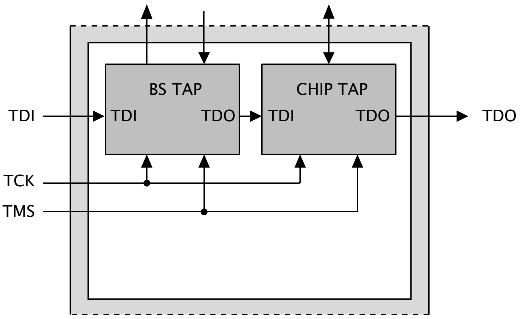


Figure 14: JTAG chain structure

The JTAG chain structure is illustrated in Figure 14. 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 15.

Figure 15: IDCODE return value

<table>
<tr><td colspan="4">Bit31</td><td colspan="24">Device Identification Register</td><td colspan="4">Bit0</td></tr>
<tr><td colspan="4">Version</td><td colspan="16">Part Number</td><td colspan="11">Manufacturer Identity</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">0</td><td colspan="4">6</td><td colspan="4">6</td><td colspan="4">3</td><td colspan="4">3</td></tr>
</table>

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 16. The OTP User ID field is read from bits [22:31] of the security register on xCORE Tile 0, *see* §9.1 (all zero on unprogrammed devices).

Figure 16: USERCODE return value

Bit31																Usercode Register															Bit0
OTP User ID										Unused				Silicon Revision																	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0				0				0				2				8				0				0				0			

13 Board Integration

The device has the following power supply pins:

- VDD pins for the xCORE Tile
- VDDIO pins for the I/O lines
- PLL_AVDD pins for the PLL
- OTP_VCC pins for the OTP

Several pins of each type are provided to minimize the effect of inductance within the package, all of which must be connected. The power supplies must be brought up monotonically and input voltages must not exceed specification at any time.

The VDD supply must ramp from 0 V to its final value within 10 ms to ensure correct startup.

The VDDIO and OTP_VCC supply must ramp to its final value before VDD reaches 0.4 V.

The PLLVDD supply should be separated from the other noisier supplies on the board. The PLL requires a very clean power supply, and a low pass filter (for example, a 4.7 Ω resistor and multi-layer ceramic capacitor) is recommended on this pin.

The following ground pins are provided:

- PLL_AGND for PLL_AVDD
- GND for all other supplies

All ground pins must be connected directly to the board ground.

The VDD and VDDIO supplies should be decoupled close to the chip by several 100 nF low inductance multi-layer ceramic capacitors between the supplies and GND (for example, 4x100nF 0402 low inductance MLCCs per supply rail). The ground side of the decoupling capacitors should have as short a path back to the GND pins as possible. A bulk decoupling capacitor of at least 10 uF should be placed on each of these supplies.

RST_N is an active-low asynchronous-assertion global reset signal. Following a reset, the PLL re-establishes lock after which the device boots up according to the boot mode (*see* §8). RST_N and must be asserted low during and after power up for 100 ns.

13.1 USB connections

USB_VBUS should be connected to the VBUS pin of the USB connector. A 2.2 uF capacitor to ground is required on the VBUS pin. A ferrite bead may be used to reduce HF noise.

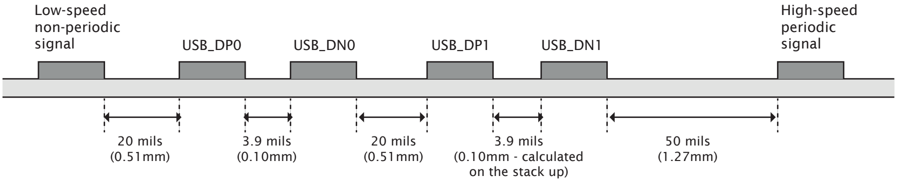
For self-powered systems, a bleeder resistor may be required to stop VBUS from floating when no USB cable is attached.

USB_DP and USB_DN should be connected to the USB connector. USB_ID does not need to be connected.

13.2 USB signal routing and placement

The USB_DP and USB_DN lines are the positive and negative data polarities of a high speed USB signal respectively. Their high-speed differential nature implies that they must be coupled and properly isolated. The board design must ensure that the board traces for USB_DP and USB_DN are tightly matched. In addition, according to the USB 2.0 specification, the USB_DP and USB_DN differential impedance must be 90 Ω.

Figure 17: USB trace separation showing a low speed signal, two differential pairs and a high-speed clock


Low-speed non-periodic signal
USB_DP0
USB_DN0
USB_DP1
USB_DN1
High-speed periodic signal
20 mils (0.51mm)
3.9 mils (0.10mm)
20 mils (0.51mm)
3.9 mils (0.10mm - calculated on the stack up)
50 mils (1.27mm)

13.2.1 General routing and placement guidelines

The following guidelines will help to avoid signal quality and EMI problems on high speed USB designs. They relate to a four-layer (Signal, GND, Power, Signal) PCB.

For best results, most of the routing should be done on the top layer (assuming the USB connector and XS2-UEF32A-1024-FB374 are on the top layer) closest to GND. Reference planes should be below the transmission lines in order to maintain control of the trace impedance.