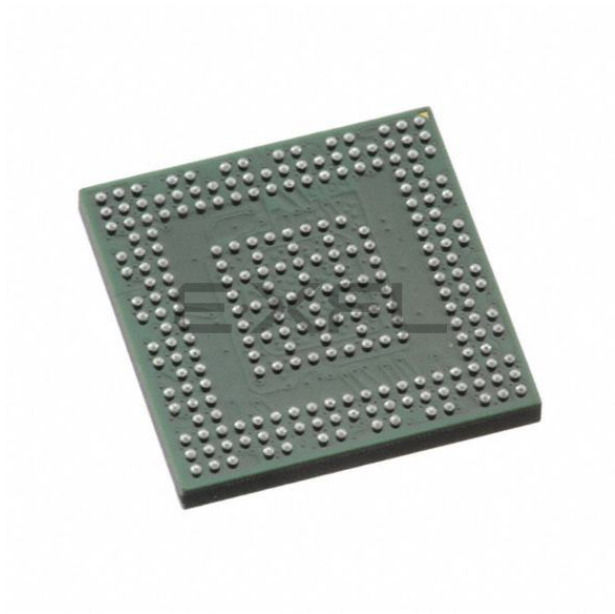


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Active
Core Processor	XCore
Core Size	32-Bit 10-Core
Speed	2000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	104
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	236-LFBGA
Supplier Device Package	236-FBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xu210-512-fb236-c20

Table of Contents

I/O pins (128)								
Signal	**Function**						**Type**	**Properties**
X0D00		$1A^0$					I/O	IOL, PD
X0D01	$X_0L3^2_{out}$	$1B^0$					I/O	IOL, PD
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	$X_0L3^3_{out}$	$1C^0$					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
X0D16	$X_0L4^4_{in}$		$4D^0$	$8B^2$	$16A^{10}$		I/O	IOR, PD
X0D17	$X_0L4^3_{in}$		$4D^1$	$8B^3$	$16A^{11}$		I/O	IOR, PD
X0D18	$X_0L4^2_{in}$		$4D^2$	$8B^4$	$16A^{12}$		I/O	IOR, PD
X0D19	$X_0L4^1_{in}$		$4D^3$	$8B^5$	$16A^{13}$		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
X0D24	$X_0L7^0_{in}$	$1I^0$					I/O	IOR, PD
X0D25	$X_0L7^0_{out}$	$1J^0$					I/O	IOR, PD
X0D26	$X_0L7^3_{out}$		$4E^0$	$8C^0$	$16B^0$		I/O	IOR, PD
X0D27	$X_0L7^4_{out}$		$4E^1$	$8C^1$	$16B^1$		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
X0D34	$X_0L7^1_{out}$	$1K^0$					I/O	IOR, PD
X0D35	$X_0L7^2_{out}$	$1L^0$					I/O	IOR, PD
X0D36		$1M^0$		$8D^0$	$16B^8$		I/O	IOL, PD
X0D37	$X_0L0^4_{in}$	$1N^0$		$8D^1$	$16B^9$		I/O	IOL, PD
X0D38	$X_0L0^3_{in}$	$1O^0$		$8D^2$	$16B^{10}$		I/O	IOL, PD
X0D39	$X_0L0^2_{in}$	$1P^0$		$8D^3$	$16B^{11}$		I/O	IOL, PD
X0D40	$X_0L0^1_{in}$			$8D^4$	$16B^{12}$		I/O	IOL, PD

(continued)

Signal	Function					Type	Properties
X0D41	$X_0L0^0_{in}$		8D[5]	16B[13]		I/O	IOL, PD
X0D42	$X_0L0^0_{out}$		8D[6]	16B[14]		I/O	IOL, PD
X0D43	$X_0L0^1_{out}$		8D[7]	16B[15]		I/O	IOL, PD
X0D49	$X_0L5^4_{in}$				32A[0]	I/O	IOR, PD
X0D50	$X_0L5^3_{in}$				32A[1]	I/O	IOR, PD
X0D51	$X_0L5^2_{in}$				32A[2]	I/O	IOR, PD
X0D52	$X_0L5^1_{in}$				32A[3]	I/O	IOR, PD
X0D53	$X_0L5^0_{in}$				32A[4]	I/O	IOR, PD
X0D54	$X_0L5^0_{out}$				32A[5]	I/O	IOR, PD
X0D55	$X_0L5^1_{out}$				32A[6]	I/O	IOR, PD
X0D56	$X_0L5^2_{out}$				32A[7]	I/O	IOR, PD
X0D57	$X_0L5^3_{out}$				32A[8]	I/O	IOR, PD
X0D58	$X_0L5^4_{out}$				32A[9]	I/O	IOR, PD
X0D61	$X_0L6^4_{in}$				32A[10]	I/O	IOR, PD
X0D62	$X_0L6^3_{in}$				32A[11]	I/O	IOR, PD
X0D63	$X_0L6^2_{in}$				32A[12]	I/O	IOR, PD
X0D64	$X_0L6^1_{in}$				32A[13]	I/O	IOR, PD
X0D65	$X_0L6^0_{in}$				32A[14]	I/O	IOR, PD
X0D66	$X_0L6^0_{out}$				32A[15]	I/O	IOR, PD
X0D67	$X_0L6^1_{out}$				32A[16]	I/O	IOR, PD
X0D68	$X_0L6^2_{out}$				32A[17]	I/O	IOR, PD
X0D69	$X_0L6^3_{out}$				32A[18]	I/O	IOR, PD
X0D70	$X_0L6^4_{out}$				32A[19]	I/O	IOR, PD
X1D00	$X_0L7^2_{in}$	1A[0]				I/O	IOR, PD
X1D01	$X_0L7^1_{in}$	1B[0]				I/O	IOR, PD
X1D02	$X_0L4^0_{in}$		4A[0]	8A[0]	16A[0]	32A[20]	I/O	IOR, PD
X1D03	$X_0L4^0_{out}$		4A[1]	8A[1]	16A[1]	32A[21]	I/O	IOR, PD
X1D04	$X_0L4^1_{out}$		4B[0]	8A[2]	16A[2]	32A[22]	I/O	IOR, PD
X1D05	$X_0L4^2_{out}$		4B[1]	8A[3]	16A[3]	32A[23]	I/O	IOR, PD
X1D06	$X_0L4^3_{out}$		4B[2]	8A[4]	16A[4]	32A[24]	I/O	IOR, PD
X1D07	$X_0L4^4_{out}$		4B[3]	8A[5]	16A[5]	32A[25]	I/O	IOR, PD
X1D08	$X_0L7^4_{in}$		4A[2]	8A[6]	16A[6]	32A[26]	I/O	IOR, PD
X1D09	$X_0L7^3_{in}$		4A[3]	8A[7]	16A[7]	32A[27]	I/O	IOR, 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	$X_0L3^1_{in}$		4D[0]	8B[2]	16A[10]		I/O	IOL, PD
X1D17	$X_0L3^0_{in}$		4D[1]	8B[3]	16A[11]		I/O	IOL, PD
X1D18	$X_0L3^0_{out}$		4D[2]	8B[4]	16A[12]		I/O	IOL, PD
X1D19	$X_0L3^1_{out}$		4D[3]	8B[5]	16A[13]		I/O	IOL, PD

(continued)

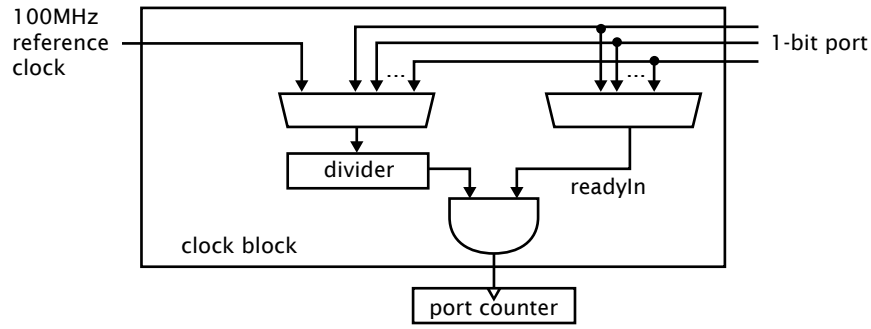


Figure 5: Clock block diagram

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 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.

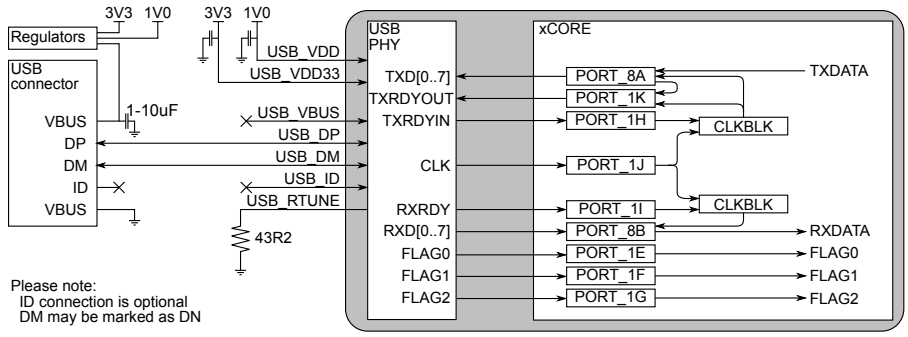


Figure 14: Bus powered *USB-device*

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

10.1 USB VBUS

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

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

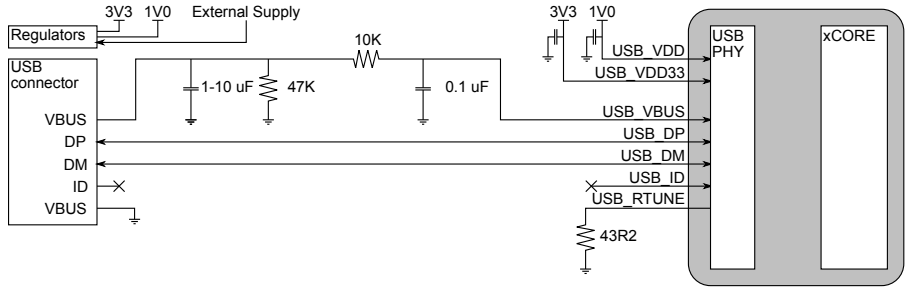


Figure 15: Self powered *USB-device*

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

The JTAG device identification register can be read by using the IDCODE instruction. Its contents are specified in Figure 17.

Figure 17: 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			

The JTAG usercode register can be read by using the USERCODE instruction. Its contents are specified in Figure 18. 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 18: 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			

12 Board Integration

The device has the following power supply pins:

- VDD pins for the xCORE Tile, including a USB_VDD pin that powers the USB PHY
- VDDIO pins for the I/O lines. Separate I/O supplies are provided for the left, top, and right side of the package; different I/O voltages may be supplied on those. The signal description (Section 4) specifies which I/O is powered from which power-supply
- PLL_AVDD pins for the PLL
- OTP_VCC pins for the OTP
- A USB_VDD33 pin for the analogue supply to the USB-PHY

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.

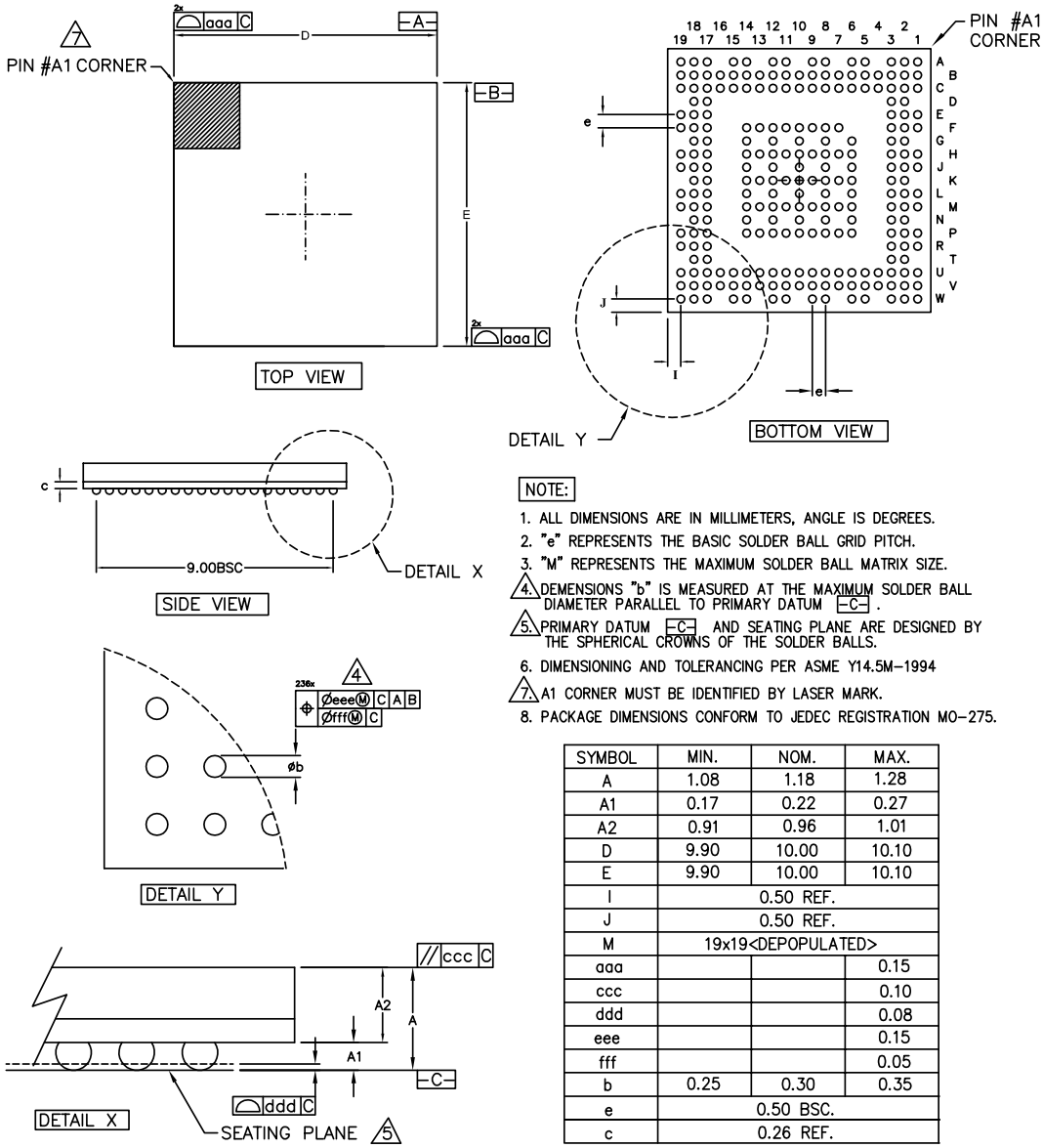
The VDD supply must ramp from 0 V to its final value within 10 ms to ensure correct startup.

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

The PLL_AVDD 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 100 nF multi-layer ceramic capacitor) is recommended on this pin.

The following ground pins are provided:

14 Package Information



NOTE:

1. ALL DIMENSIONS ARE IN MILLIMETERS, ANGLE IS DEGREES.
2. "e" REPRESENTS THE BASIC SOLDER BALL GRID PITCH.
3. "M" REPRESENTS THE MAXIMUM SOLDER BALL MATRIX SIZE.
4. DEMENSIONS "b" IS MEASURED AT THE MAXIMUM SOLDER BALL DIAMETER PARALLEL TO PRIMARY DATUM -C- .
5. PRIMARY DATUM -C- AND SEATING PLANE ARE DESIGNED BY THE SPHERICAL CROWNS OF THE SOLDER BALLS.
6. DIMENSIONING AND TOLERANCING PER ASME Y14.5M–1994
7. A1 CORNER MUST BE IDENTIFIED BY LASER MARK.
8. PACKAGE DIMENSIONS CONFORM TO JEDEC REGISTRATION MO–275.

SYMBOL	MIN.	NOM.	MAX.
A	1.08	1.18	1.28
A1	0.17	0.22	0.27
A2	0.91	0.96	1.01
D	9.90	10.00	10.10
E	9.90	10.00	10.10
I		0.50 REF.	
J		0.50 REF.	
M		19x19<DEPOPULATED>	
aaa			0.15
ccc			0.10
ddd			0.08
eee			0.15
fff			0.05
b	0.25	0.30	0.35
e		0.50 BSC.	
c		0.26 REF.	

14.1 Part Marking

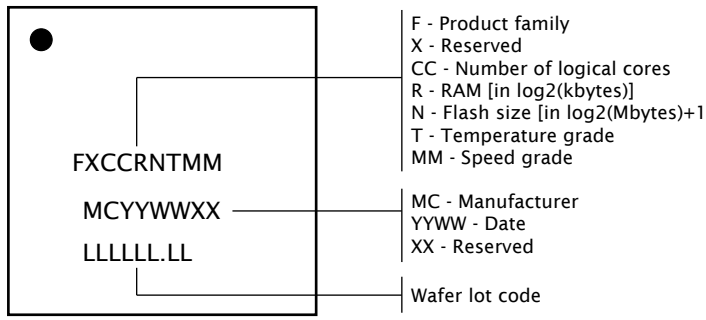


Figure 31: Part marking scheme

15 Ordering Information

Figure 32: Orderable part numbers

Product Code	Marking	Qualification	Speed Grade
XU210-512-FB236-C20	U11090C20	Commercial	1000 MIPS
XU210-512-FB236-I20	U11090I20	Industrial	1000 MIPS

A write message comprises the following:

control-token 36	24-bit response channel-end identifier	8-bit register number	8-bit size	data	control-token 1

The response to a write message comprises either control tokens 3 and 1 (for success), or control tokens 4 and 1 (for failure).

A read message comprises the following:

control-token 37	24-bit response channel-end identifier	8-bit register number	8-bit size	control-token 1

The response to the read message comprises either control token 3, data, and control-token 1 (for success), or control tokens 4 and 1 (for failure).

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.

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.

D Node Configuration

The digital node control registers can be accessed using configuration reads and writes (use `write_node_config_reg(device, ...)` and `read_node_config_reg(device, ...)` for reads and writes).

Number	Perm	Description
0x00	RO	Device identification
0x01	RO	System switch description
0x04	RW	Switch configuration
0x05	RW	Switch node identifier
0x06	RW	PLL settings
0x07	RW	System switch clock divider
0x08	RW	Reference clock
0x09	R	System JTAG device ID register
0x0A	R	System USERCODE register
0x0C	RW	Directions 0-7
0x0D	RW	Directions 8-15
0x10	RW	DEBUG_N configuration, tile 0
0x11	RW	DEBUG_N configuration, tile 1
0x1F	RO	Debug source
0x20 .. 0x28	RW	Link status, direction, and network
0x40 .. 0x47	RO	PLink status and network
0x80 .. 0x88	RW	Link configuration and initialization
0xA0 .. 0xA7	RW	Static link configuration

Figure 36: Summary

D.1 Device identification: 0x00

This register contains version and revision identifiers and the mode-pins as sampled at boot-time.

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Sampled values of BootCtl pins on Power On Reset.
15:8	RO		SSwitch revision.
7:0	RO		SSwitch version.

0x00: Device identification

D.2 System switch description: 0x01

This register specifies the number of processors and links that are connected to this switch.

0x01: System switch description

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Number of SLinks on the SSwitch.
15:8	RO		Number of processors on the SSwitch.
7:0	RO		Number of processors on the device.

D.3 Switch configuration: 0x04

This register enables the setting of two security modes (that disable updates to the PLL or any other registers) and the header-mode.

0x04: Switch configuration

Bits	Perm	Init	Description
31	RW	0	0 = SSCTL registers have write access. 1 = SSCTL registers can not be written to.
30:9	RO	-	Reserved
8	RW	0	0 = PLL_CTL_REG has write access. 1 = PLL_CTL_REG can not be written to.
7:1	RO	-	Reserved
0	RW	0	0 = 2-byte headers, 1 = 1-byte headers (reset as 0).

D.4 Switch node identifier: 0x05

This register contains the node identifier.

0x05: Switch node identifier

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	The unique ID of this node.

D.5 PLL settings: 0x06

An on-chip PLL multiplies the input clock up to a higher frequency clock, used to clock the I/O, processor, and switch, see Oscillator. Note: a write to this register will cause the tile to be reset.

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.

0x2C: UIFM PID

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

F.13 UIFM Endpoint: 0x30

The last endpoint seen

0x30: UIFM Endpoint

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

F.14 UIFM Endpoint match: 0x34

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

0x34: UIFM Endpoint match

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

F.15 OTG Flags mask: 0x38

0x38: OTG Flags mask

Bits	Perm	Init	Description
31:0	RW	0	Data

F.16 UIFM power signalling: 0x3C

0x3C: UIFM power signalling

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

G JTAG, xSCOPE and Debugging

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

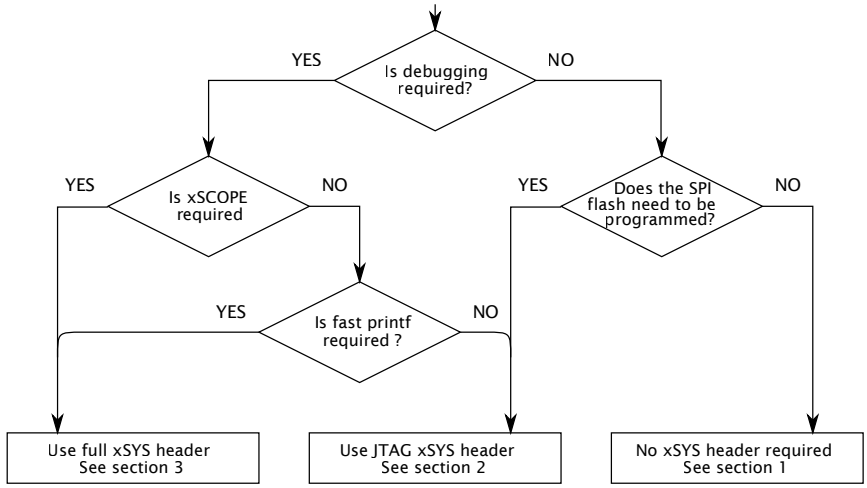


Figure 39: Decision diagram for the xSYS header

G.1 No xSYS header

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

G.2 JTAG-only xSYS header

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

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

- TDI to pin 5 of the xSYS header
- TMS to pin 7 of the xSYS header
- TCK to pin 9 of the xSYS header
- DEBUG_N to pin 11 of the xSYS header

- TDO to pin 13 of the xSYS header

The RST_N net should be open-drain, active-low, and have a pull-up to VDDIO.

G.3 Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section G.2, and then connect a 2-wire xCONNECT Link to the xSYS header. The links can be found in the Signal description table (Section 4): they are labelled XL0, XL1, etc in the function column. The 2-wire link comprises two inputs and outputs, labelled ${}^{1}_{out}$, ${}^{0}_{out}$, ${}^{0}_{in}$, and ${}^{1}_{in}$. For example, if you choose to use XL0 for xSCOPE I/O, you need to connect up $XL0^{1}_{out}$, $XL0^{0}_{out}$, $XL0^{0}_{in}$, $XL0^{1}_{in}$ as follows:

- $XL0^{1}_{out}$ (X0D43) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- $XL0^{0}_{out}$ (X0D42) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- $XL0^{0}_{in}$ (X0D41) to pin 14 of the xSYS header.
- $XL0^{1}_{in}$ (X0D40) to pin 18 of the xSYS header.

H Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XU210-512-FB236. Each of the following sections contains items to check for each design.

H.1 Power supplies

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

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

☐ The VDD (core) supply is capable of supplying 700 mA (Section 12 and Figure 22).

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

H.2 Power supply decoupling

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

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

H.3 Power on reset

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

H.4 Clock

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

☐ 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.