



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

### Details

Product Status	Active
Core Processor	XCore
Core Size	32-Bit 24-Core
Speed	4000MIPS
Connectivity	-
Peripherals	-
Number of I/O	176
Program Memory Size	-
Program Memory Type	ROMIess
EEPROM Size	-
RAM Size	512K x 8
Voltage - Supply (Vcc/Vdd)	0.95V ~ 3.6V
Data Converters	-
Oscillator Type	External
Operating Temperature	-40°C ~ 85°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/xl224-512-fb374-i40

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong

# 1 xCORE Multicore Microcontrollers

The xCORE200 Series is a comprehensive range of 32-bit multicore microcontrollers that brings the low latency and timing determinism of the xCORE architecture to mainstream embedded applications. Unlike conventional microcontrollers, xCORE multicore microcontrollers execute multiple real-time tasks simultaneously and communicate between tasks using a high speed network. Because xCORE multicore microcontrollers are completely deterministic, you can write software to implement functions that traditionally require dedicated hardware.



Figure 1: XL224-512-FB374 block diagram

Key features of the XL224-512-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
- ▶ JTAG The JTAG module can be used for loading programs, boundary scan testing, in-circuit source-level debugging and programming the OTP memory. Section 10

#### 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.



ports are available. All pins of a port provide either output or input. Signals in different directions cannot be mapped onto the same port.

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.



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-L 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	MC	DDE	Tile	PLL Ratio	PLL setting		gs
_	Frequency	1	0	Frequency		OD	F	R
7:	3.25-10 MHz	0	0	130-400 MHz	40	1	159	0
ier	9-25 MHz	1	1	144-400 MHz	16	1	63	0
nd	25-50 MHz	1	0	167-400 MHz	8	1	31	0
ns	50-100 MHz	0	1	196-400 MHz	4	1	15	0

-XMOS

XS2-L24A-512-FB374

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  $0 \le R \le 63$ ,  $0 \le F \le 4095$ ,  $0 \le OD \le 7$ , and  $260MHz \le F_{osc} \times \frac{F+1}{2} \times \frac{1}{R+1} \le 1.3GHz$ . 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 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. When the device is taken out of reset by releasing RST\_N the processor starts its internal reset process. After 15-150  $\mu$ s (depending on the input clock) the processor boots.

Pin X2D06 must be pulled high with an external pull-up whilst the chip comes out of reset, to ensure that tile 2 will boot from link. X2D04, X2D05, and X2D07 should be kept low whilst the chip comes out of reset.

The xCORE Tile boot procedure is illustrated in Figure 8. If bit 5 of the security register (*see* §9.1) is set, the device boots from OTP. To get a high value, a 3K3 pull-up resistor should be strapped onto the pin. To assure a low value, a pull-down resistor is required if other external devices are connected to this port.



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.

### 11.1 Land patterns and solder stencils

The package is a 374 ball Fine Ball Grid Array (FBGA) on a 0.8 mm pitch.

The land patterns and solder stencils will depend on the PCB manufacturing process. We recommend you design them with using the IPC specifications *"Generic Requirements for Surface Mount Design and Land Pattern Standards"* IPC-7351B. This standard aims to achieve desired targets of heel, toe and side fillets for solder-joints. The mechanical drawings in Section 13 specify the dimensions and tolerances.

### 11.2 Ground and Thermal Vias

Vias next to each ground ball into the ground plane of the PCB are recommended for a low inductance ground connection and good thermal performance.

### 11.3 Moisture Sensitivity

XMOS devices are, like all semiconductor devices, susceptible to moisture absorption. When removed from the sealed packaging, the devices slowly absorb moisture from the surrounding environment. If the level of moisture present in the device is too high during reflow, damage can occur due to the increased internal vapour pressure of moisture. Example damage can include bond wire damage, die lifting, internal or external package cracks and/or delamination.

All XMOS devices are Moisture Sensitivity Level (MSL) 3 - devices have a shelf life of 168 hours between removal from the packaging and reflow, provided they are stored below 30C and 60% RH. If devices have exceeded these values or an included moisture indicator card shows excessive levels of moisture, then the parts should be baked as appropriate before use. This is based on information from *Joint IPC/JEDEC Standard For Moisture/Reflow Sensitivity Classification For Nonhermetic Solid State Surface-Mount Devices* J-STD-020 Revision D.

-XM()S

XS2-L24A-512-FB374

24

# 12 DC and Switching Characteristics

Symbol	Parameter	MIN	ТҮР	MAX	UNITS	Notes
VDD	Tile DC supply voltage	0.95	1.00	1.05	V	
VDDIO	I/O supply voltage	2.30	3.30	3.60	V	
VDDIOT 3v3	I/O supply voltage	3.135	3.30	3.465	V	
VDDIOT 2v5	I/O supply voltage	2.375	2.50	2.625	V	
USB_VDD	USB tile DC supply voltage	0.95	1.00	1.05	V	
VDD33	Peripheral supply	3.135	3.30	3.465	V	
PLL_AVDD	PLL analog supply	0.95	1.00	1.05	V	
CI	xCORE Tile I/O load capacitance			25	pF	
Та	Ambient operating temperature (Commercial)	0		70	°C	
	Ambient operating temperature (Industrial)	-40		85	°C	
Tj	Junction temperature			125	°C	
Tstg	Storage temperature	-65		150	°C	

### 12.1 Operating Conditions

Figure 17: Operating conditions

## 12.2 DC Characteristics, VDDIO=3V3

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	А
V(IL)	Input low voltage	-0.30		0.70	V	А
V(OH)	Output high voltage	2.20			V	B, C
V(OL)	Output low voltage			0.40	V	B, C
I(PU)	Internal pull-up current (Vin=0V)	-100			μA	D
I(PD)	Internal pull-down current (Vin=3.3V)			100	μA	D
I(LC)	Input leakage current	-10		10	μA	

Figure 18: DC characteristics

A All pins except power supply pins.

B Pins X1D40, X1D41, X1D42, X1D43, X1D26, X1D27, X3D40, X3D41, X3D42, X3D43, X3D26, and X3D27 are nominal 8 mA drivers, the remainder of the general-purpose I/Os are 4 mA.

C Measured with 4 mA drivers sourcing 4 mA, 8 mA drivers sourcing 8 mA.

D Used to guarantee logic state for an I/O when high impedance. The internal pull-ups/pull-downs should not be used to pull external circuitry. In order to pull the pin to the opposite state, a 4K7 resistor is recommended to overome the internal pull current.



X010434,

	Symbol	Parameter	MIN	ТҮР	MAX	UNITS	Notes
Figure 24: I/O AC char- acteristics	T(XOVALID)	Input data valid window	8			ns	
	T(XOINVALID)	Output data invalid window	9			ns	
	T(XIFMAX)	Rate at which data can be sampled with respect to an external clock			60	MHz	

#### 12.7 xCORE Tile I/O AC Characteristics

The input valid window parameter relates to the capability of the device to capture data input to the chip with respect to an external clock source. It is calculated as the sum of the input setup time and input hold time with respect to the external clock as measured at the pins. The output invalid window specifies the time for which an output is invalid with respect to the external clock. Note that these parameters are specified as a window rather than absolute numbers since the device provides functionality to delay the incoming clock with respect to the incoming data.

Information on interfacing to high-speed synchronous interfaces can be found in the XS1 Port I/O Timing document, X5821.

#### 12.8 xConnect Link Performance

	Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
	B(2blinkP)	2b link bandwidth (packetized)			87	MBit/s	А, В
Figure 25:	B(5blinkP)	5b link bandwidth (packetized)			217	MBit/s	А, В
Link	B(2blinkS)	2b link bandwidth (streaming)			100	MBit/s	В
performance	B(5blinkS)	5b link bandwidth (streaming)			250	MBit/s	В

A Assumes 32-byte packet in 3-byte header mode. Actual performance depends on size of the header and payload.

B 7.5 ns symbol time.

The asynchronous nature of links means that the relative phasing of CLK clocks is not important in a multi-clock system, providing each meets the required stability criteria.

12.9	ITAG	Timina
12.5	JIAG	1 mmg

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
f(TCK_D)	TCK frequency (debug)			18	MHz	
f(TCK_B)	TCK frequency (boundary scan)			10	MHz	
T(SETUP)	TDO to TCK setup time	5			ns	А
T(HOLD)	TDO to TCK hold time	5			ns	А
T(DELAY)	TCK to output delay			15	ns	В

Figure 26: JTAG timing

A Timing applies to TMS and TDI inputs.

B Timing applies to TDO output from negative edge of TCK.

All JTAG operations are synchronous to TCK apart from the global asynchronous reset TRST\_N.

-XM()S

## 13.1 Part Marking



# 14 Ordering Information

Figure 28:	Product Code	Marking	Qualification	Speed Grade
Orderable	XL224-512-FB374-C40	L12490C40	Commercial	2000 MIPS
part numbers	XL224-512-FB374-I40	L12490C40	Industrial	2000 MIPS

# **B** Processor Status Configuration

The processor status control registers can be accessed directly by the processor using processor status reads and writes (use getps(reg) and setps(reg,value) for reads and writes).

Number	Perm	Description	
0x00	RW	RAM base address	
0x01	RW	Vector base address	
0x02	RW	xCORE Tile control	
0x03	RO	xCORE Tile boot status	
0x05	RW	Security configuration	
0x06	RW	Ring Oscillator Control	
0x07	RO	Ring Oscillator Value	
0x08	RO	Ring Oscillator Value	
0x09	RO	Ring Oscillator Value	
0x0A	RO	Ring Oscillator Value	
0x0C	RO	RAM size	
0x10	DRW	Debug SSR	
0x11	DRW	Debug SPC	
0x12	DRW	Debug SSP	
0x13	DRW	DGETREG operand 1	
0x14	DRW	DGETREG operand 2	
0x15	DRW	Debug interrupt type	
0x16	DRW	Debug interrupt data	
0x18	DRW	Debug core control	
0x20 0x27	DRW	Debug scratch	
0x30 0x33	DRW	Instruction breakpoint address	
0x40 0x43	DRW	Instruction breakpoint control	
0x50 0x53	DRW	Data watchpoint address 1	
0x60 0x63	DRW	Data watchpoint address 2	
0x70 0x73	DRW	Data breakpoint control register	
0x80 0x83	DRW	Resources breakpoint mask	
0x90 0x93	DRW	Resources breakpoint value	
0x9C 0x9F	DRW	Resources breakpoint control register	

Figure 30:

Summary

X010434,

0x13	Bits	Perm	Init	Description
DGETREG	31:8	RO	-	Reserved
operand 1	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

4.	Bits	Perm	Init	Description
ч. G	31:5	RO	-	Reserved
2	4:0	DRW		Register number to be read

### B.17 Debug interrupt type: 0x15

Register that specifies what activated the debug interrupt.

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

**0x15:** Debug interrupt type

# 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.

-XMOS-

39

# C Tile Configuration

The xCORE Tile control registers can be accessed using configuration reads and writes (use write\_tile\_config\_reg(tileref, ...) and read\_tile\_config\_reg(tileref,  $\rightarrow$  ...) for reads and writes).

Number	Perm	Description
0x00	CRO	Device identification
0x01	CRO	xCORE Tile description 1
0x02	CRO	xCORE Tile description 2
0x04	CRW	Control PSwitch permissions to debug registers
0x05	CRW	Cause debug interrupts
0x06	CRW	xCORE Tile clock divider
0x07	CRO	Security configuration
0x20 0x27	CRW	Debug scratch
0x40	CRO	PC of logical core 0
0x41	CRO	PC of logical core 1
0x42	CRO	PC of logical core 2
0x43	CRO	PC of logical core 3
0x44	CRO	PC of logical core 4
0x45	CRO	PC of logical core 5
0x46	CRO	PC of logical core 6
0x47	CRO	PC of logical core 7
0x60	CRO	SR of logical core 0
0x61	CRO	SR of logical core 1
0x62	CRO	SR of logical core 2
0x63	CRO	SR of logical core 3
0x64	CRO	SR of logical core 4
0x65	CRO	SR of logical core 5
0x66	CRO	SR of logical core 6
0x67	CRO	SR of logical core 7

Figure 31: Summary

# C.1 Device identification: 0x00

This register identifies the xCORE Tile

-XMOS

Bits

31:0

**0x41:** PC of logical core 1

Perm	Init	Description		
CRO		Value.		

## C.11 PC of logical core 2: 0x42

Value of the PC of logical core 2.

0x42: PC of logical core 2

Bits	Perm	Init	Description
31:0	CRO		Value.

## C.12 PC of logical core 3: 0x43

Value of the PC of logical core 3.

0x42				
PC of logical	Bits	Perm	Init	Description
core 3	31:0	CRO		Value.

## C.13 PC of logical core 4: 0x44

Value of the PC of logical core 4.

**0x44** PC of logical core 4

0x44: ogical	Bits	Perm	Init	Description
ore 4	31:0	CRO		Value.

# C.14 PC of logical core 5: 0x45

Value of the PC of logical core 5.

**0x45:** PC of logical core 5

Bits	Perm	Init	Description	
31:0	CRO		Value.	

# **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,  $\rightarrow$  ...) 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 32: 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
<b>0x00:</b> Device ification	23:16	RO		Sampled values of BootCtl pins on Power On Reset.
	15:8	RO		SSwitch revision.
	7:0	RO		SSwitch version.

-XMOS<sup>®</sup>

identi

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.	

**0x1F:** Debug source

## 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.

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.

0x20 .. 0x28: Link status, direction, and network

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

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

## D.18 Static link configuration: 0xA0 .. 0xA7

These registers are used for static (ie, non-routed) links. When a link is made static, all traffic is forwarded to the designated channel end and no routing is attempted. The registers control links C, D, A, B, G, H, E, and F in that order.

Bits	Perm	Init	Description	
31	RW	0	Enable static forwarding.	
30:9	RO	-	Reserved	
8	RW	0	The destination processor on this node that packets received in static mode are forwarded to.	
7:5	RO	-	Reserved	
4:0	RW	0	The destination channel end on this node that packets received in static mode are forwarded to.	

**0xA0 .. 0xA7:** Static link configuration



#### F.5 Boot

- □ The device is connected to a QSPI flash for booting, connected to X0D01, X0D04..X0D07, and X0D10 (Section 8). If not, you must boot the device through OTP or JTAG, or set it to boot from SPI and connect a SPI flash.
- ☐ The Flash that you have chosen is supported by **xflash**, or you have created a specification file for it.

### F.6 JTAG, XScope, and debugging

- $\Box$  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 high and low appropriately (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.

- $\Box$  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).

X010434,

# G PCB Layout Design Check List

This section is a checklist for use by PCB designers using the XS2-L24A-512-FB374. Each of the following sections contains items to check for each design.

### G.1 Ground Plane

- Each ground ball has a via to minimize impedance and conduct heat away from the device. (Section 11.2)
- Other than ground vias, there are no (or only a few) vias underneath or closely around the device. This create a good, solid, ground plane.

### G.2 Power supply decoupling

- $\Box$  The decoupling capacitors are all placed close to a supply pin (Section 11).
- $\Box$  The decoupling capacitors are spaced around the device (Section 11).
- The ground side of each decoupling capacitor has a direct path back to the center ground of the device.

### G.3 PLL\_AVDD

The PLL\_AVDD filter (especially the capacitor) is placed close to the PLL\_AVDD pin (Section 11).

# H Associated Design Documentation

Document Title	Information	Document Number
Estimating Power Consumption For XS1-L Devices	Power consumption	X4271
Programming XC on XMOS Devices	Timers, ports, clocks, cores and channels	X9577
xTIMEcomposer User Guide	Compilers, assembler and linker/mapper	X3766
	Timing analyzer, xScope, debugger	
	Flash and OTP programming utilities	

# I Related Documentation

Document Title	Information	Document Number
The XMOS XS1 Architecture	ISA manual	X7879
XS1 Port I/O Timing	Port timings	X5821
xCONNECT Architecture	Link, switch and system information	X4249
XS1-L Link Performance and Design Guidelines	Link timings	X2999
XS1-L Clock Frequency Control	Advanced clock control	X1433
XS1-L Active Power Conservation	Low-power mode during idle	X7411

-XMOS<sup>®</sup>-

65