Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	XCore
Core Size	32-Bit 6-Core
Speed	400MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	42
Program Memory Size	64KB (16K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
Voltage - Supply (Vcc/Vdd)	0.90V ~ 5.5V
Data Converters	A/D 4x12b
Oscillator Type	Internal
Operating Temperature	0°C ~ 70°C (TA)
Mounting Type	Surface Mount
Package / Case	96-LFBGA
Supplier Device Package	96-FBGA (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xs1-a6a-64-fb96-c4

1 xCORE Multicore Microcontrollers

The XS1-A 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.

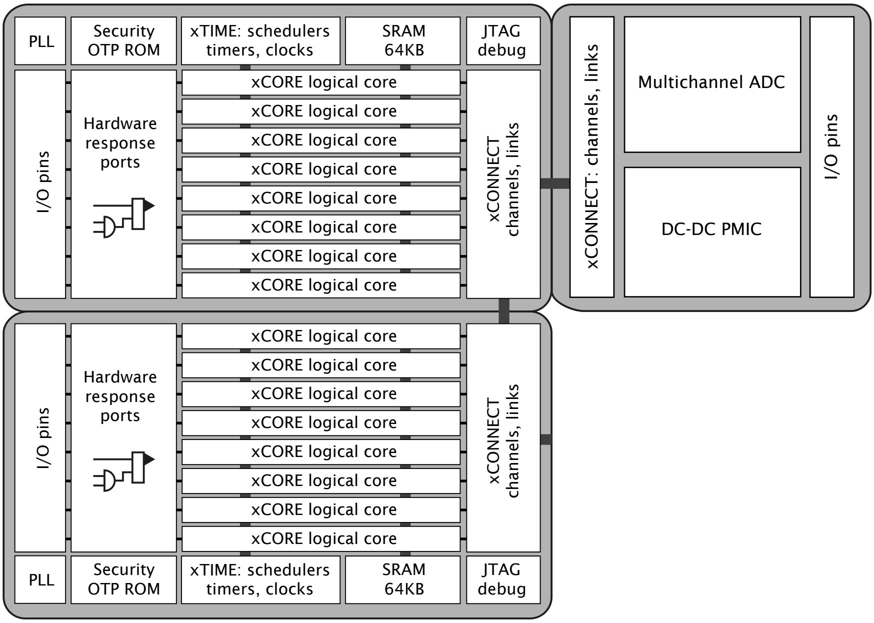

Figure 1: XS1-A Series:6-16 core devices

Key features of the XS1-A6A-64-FB96 include:

- **Tiles**: Devices consist of one or more xCORE tiles. Each tile contains between four 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 7.1
- **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 7.2

2 XS1-A6A-64-FB96 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Six real-time logical cores
  - Core share up to 500 MIPS
  - Each logical core has:
    - Guaranteed throughput of between 1/4 and 1/6 of tile MIPS
    - 16x32bit dedicated registers
  - 159 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
- **12b 1MSPS 4-channel SAR Analog-to-Digital Converter**
- **1 x LDO**
- **2 x DC-DC converters and Power Management Unit**
- **Watchdog Timer**
- **Onchip clocks/oscillators**
  - Crystal oscillator
  - 20MHz/31kHz silicon oscillators
- **Programmable I/O**
  - 42 general-purpose I/O pins, configurable as input or output
    - Up to 16 x 1bit port, 6 x 4bit port, 3 x 8bit port, 1 x 16bit port
    - 2 xCONNECT links
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 32 channel ends for communication with other cores, on or off-chip
- **Memory**
  - 64KB internal single-cycle SRAM for code and data storage
  - 8KB internal OTP for application boot code
  - 128 bytes Deep Sleep Memory
- **Hardware resources**
  - 6 clock blocks
  - 10 timers
  - 4 locks
- **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**
  - 5: 500 MIPS
  - 4: 400 MIPS
- **Power Consumption (typical)**
  - 300 mW at 500 MHz (typical)
  - Sleep Mode: 500 µW
- **96-pin FBGA package 0.8 mm pitch**

7 xCORE Tile Resources

7.1 Logical cores

The tile has 6 active logical cores, which issue instructions down a shared four-stage pipeline. Instructions from the active cores are issued round-robin. If up to four logical cores are active, each core is allocated a quarter of the processing cycles. If more than four logical cores are active, each core is allocated at least $1/n$ cycles (for n cores). Figure 4 shows the guaranteed core performance depending on the number of cores used.

Figure 4: Logical core performance

Speed grade	MIPS	Frequency	Minimum MIPS per core (for *n* cores)							
			1	2	3	4	5	6		
4	400 MIPS	400 MHz	100	100	100	100	80	67		
5	500 MIPS	500 MHz	125	125	125	125	100	83		

There is no way that the performance of a logical core can be reduced below these predicted levels. 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 four 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.

7.2 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.

7.3 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 XS1-A6A-64-FB96, 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.

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.

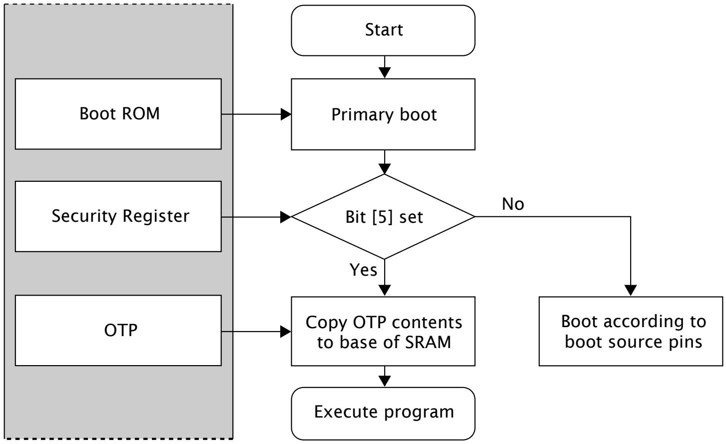

Figure 9: Boot procedure

MODE[3]	MODE[2]	Boot Source
0	0	None: Device waits to be booted via JTAG
0	1	Reserved
1	0	xConnect Link B
1	1	SPI

Figure 10: Boot source pins

- 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.

9.1 Boot from SPI master

If set to boot from SPI master, the processor enables the four pins specified in Figure 11, and drives the SPI clock at 2.5 MHz (assuming a 400 MHz core clock). A READ command is issued with a 24-bit address 0x000000. The clock polarity and phase are 0 / 0.

Pin	Signal	Description
X0D00	MISO	Master In Slave Out (Data)
X0D01	SS	Slave Select
X0D10	SCLK	Clock
X0D11	MOSI	Master Out Slave In (Data)

Figure 11: SPI master pins

systems to be partially programmed, dedicating one or more tiles to perform a particular function, leaving the other tiles user-programmable.

9.4 Security register

The security register enables security features on the xCORE tile. The features shown in Figure 12 provide a strong level of protection and are sufficient for providing strong IP security.

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 processor is forced to boot from address 0 of the OTP, allowing the processor boot ROM to be bypassed (*see* §9).
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.
Disable Global Debug	14	Disables access to the DEBUG_N pin.
	21..15	General purpose software accessable security register available to end-users.
	31..22	General purpose user programmable JTAG UserID code extension.

Figure 12: Security register features

10 Memory

10.1 OTP

The 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.

17.3 DC2 Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
VDD1V8	1V8 Supply Voltage		1.80		V	
V(RIPPLE)	Ripple Voltage (peak to peak)		10	40	mV	
V(ACC)	Voltage Accuracy	-5		5	%	A
F(S)	Switching Frequency		1		MHz	
F(SVAR)	Variation in Switching Frequency	-10		10	%	
Effic	Efficiency		80		%	
PGT(HIGH)	Powergood Threshold (High)		95		%/VDD1V8	
PGT(LOW)	Powergood Threshold (Low)		80		%/VDD1V8	

A If supplied externally.

Figure 26: DC2 characteristics

17.4 ADC Characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
N	Resolution		12		bits	
Fs	Conversion Speed			1	MSPS	
Nch	Number of Channels		4			
Vin	Input Range	0		AVDD	V	
DNL	Differential Non Linearity	-1		1.5	LSB	
INL	Integral Non Linearity	-4		4	LSB	
E(GAIN)	Gain Error	-10		10	LSB	
E(OFFSET)	Offset Error	-3		3	mV	
T(PWRUP)	Power time for ADC Clock Fclk			7	1/Fclk	
ENOB	Effective Number of bits		10			

Figure 27: ADC characteristics

17.5 Digital I/O Characteristics

Figure 28: Digital I/O characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
V(IH)	Input high voltage	2.00		3.60	V	A
V(IL)	Input low voltage	-0.30		0.70	V	A
V(OH)	Output high voltage	2.00			V	B, C
V(OL)	Output low voltage			0.60	V	B, C
R(PU)	Pull-up resistance		35K		Ω	D
R(PD)	Pull-down resistance		35K		Ω	D

A All pins except power supply pins.

B Ports 1A, 1D, 1E, 1H, 1I, 1J, 1K and 1L 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.

17.6 ESD Stress Voltage

Figure 29: ESD stress voltage

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
HBM	Human body model			2.00	kV	
CDM	Charged Device Model			500	V	

17.7 Device Timing Characteristics

Figure 30: Device timing characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
T(RST)	Reset pulse width	5			µs	A
T(INIT)	Initialisation (On Silicon Oscillator)			TBC	ms	
	Initialisation (Crystal Oscillator)			TBC	ms	
T(WAKE)	Wake up time (Sleep to Active)			TBC	ms	
T(SLEEP)	Sleep Time (Active to Sleep)			TBC	ms	

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

17.8 Crystal Oscillator Characteristics

Figure 31: Crystal oscillator characteristics

Symbol	Parameter	MIN	TYP	MAX	UNITS	Notes
F(FO)	Input Frequency	5		30	MHz	

B.19 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.20 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.

0x30 .. 0x33: Instruction breakpoint address

Bits	Perm	Init	Description
31:0	DRW		Value.

B.21 Instruction breakpoint control: 0x40 .. 0x43

This register controls which logical cores may take an instruction breakpoint, and under which condition.

0x40 .. 0x43: Instruction breakpoint control

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each logical core in the tile allowing the breakpoint to be enabled individually for each logical core.
15:2	RO	-	Reserved
1	DRW	0	Set to 1 to cause an instruction breakpoint if the PC is not equal to the breakpoint address. By default, the breakpoint is triggered when the PC is equal to the breakpoint address.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.22 Data watchpoint address 1: 0x50 .. 0x53

This set of registers contains the first address for the four data watchpoints.

0x04: Control PSwitch permissions to debug registers

Bits	Perm	Init	Description
31:1	RO	-	Reserved
0	CRW		Set to 1 to restrict PSwitch access to all CRW marked registers to become read-only rather than read-write.

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	RO	0	Set to 1 when the processor is in debug mode.
0	CRW	0	Set to 1 to 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
31:8	RO	-	Reserved
7:0	RW		Value of the clock divider minus one.

C.7 Security configuration: 0x07

Copy of the security register as read from OTP.

0x07: Security configuration

Bits	Perm	Init	Description
31:0	RO		Value.

C.8 PLink status: 0x10 .. 0x13

Status of each of the four processor links; connecting the xCORE tile to the switch.

0x0C: Directions 0-7

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

D.9 Directions 8-15: 0x0D

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

0x0D: Directions 8-15

Bits	Perm	Init	Description
31:28	RW	0	The direction for packets whose first mismatching bit is 15.
27:24	RW	0	The direction for packets whose first mismatching bit is 14.
23:20	RW	0	The direction for packets whose first mismatching bit is 13.
19:16	RW	0	The direction for packets whose first mismatching bit is 12.
15:12	RW	0	The direction for packets whose first mismatching bit is 11.
11:8	RW	0	The direction for packets whose first mismatching bit is 10.
7:4	RW	0	The direction for packets whose first mismatching bit is 9.
3:0	RW	0	The direction for packets whose first mismatching bit is 8.

D.10 DEBUG_N configuration: 0x10

Configures the behavior of the DEBUG_N pin.

0x10: DEBUG_N configuration

Bits	Perm	Init	Description
31:2	RO	-	Reserved
1	RW	0	Set to 1 to enable signals on DEBUG_N to generate DCALL on the core.
0	RW	0	When set to 1, the DEBUG_N wire will be pulled down when the node enters debug mode.

D.13 PLink status and network: 0x40 .. 0x43

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

0x40 .. 0x43: PLink status and network

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RO		If this link is currently routing data into the switch, this field specifies the type of link that the data is routed to: 0: plink 1: external link 2: internal control link
23:16	RO	0	If the link is routing data into the switch, this field specifies the destination link number to which all tokens are sent.
15:6	RO	-	Reserved
5:4	RW	0	Determines the network to which this link belongs, set for quality of service.
3	RO	-	Reserved
2	RO	0	Set to 1 if the current packet is junk and being thrown away. A packet is considered junk if, for example, it is not routable.
1	RO	0	Set to 1 if the switch is routing data into the link, and if a route exists from another link.
0	RO	0	Set to 1 if the link is routing data into the switch, and if a route is created to another link on the switch.

D.14 Link configuration and initialization: 0x80 .. 0x87

These registers contain configuration and debugging information specific to external links. The link speed and width can be set, the link can be initialized, and the link status can be monitored. The registers control links C, D, A, B, G, H, E, and F in that order.

Bits	Perm	Init	Description
31	RW	0	Write '1' to this bit to enable the link, write '0' to disable it. This bit controls the muxing of ports with overlapping links.
30	RW	0	Set to 0 to operate in 2 wire mode or 1 to operate in 5 wire mode
29:28	RO	-	Reserved
27	RO	0	Set to 1 on error: an RX buffer overflow or illegal token encoding has been received. This bit clears on reading.
26	RO	0	1 if this end of the link has issued credit to allow the remote end to transmit.
25	RO	0	1 if this end of the link has credits to allow it to transmit.
24	WO	0	Set to 1 to initialize a half-duplex link. This clears this end of the link's credit and issues a HELLO token; the other side of the link will reply with credits. This bit is self-clearing.
23	WO	0	Set to 1 to reset the receiver. The next symbol that is detected will be assumed to be the first symbol in a token. This bit is self-clearing.
22	RO	-	Reserved
21:11	RW	0	The number of system clocks between two subsequent transitions within a token
10:0	RW	0	The number of system clocks between two subsequent transmit tokens.

0x80 .. 0x87: Link configuration and initialization

D.15 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.
30:5	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

Figure 46: Summary

Number	Perm	Description
0x00 .. 0x7F	RW	Deep sleep memory
0xFF	RW	Deep sleep memory valid

G.1 Deep sleep memory: 0x00 .. 0x7F

128 bytes of memory that can be used to hold data when the xCORE Tile is powered down.

0x00 .. 0x7F: Deep sleep memory

Bits	Perm	Init	Description
7:0	RW		User defined data

G.2 Deep sleep memory valid: 0xFF

One byte of memory that is reset to 0. The program can write a non zero value in this register to indicate that the data in deep sleep memory is valid.

0xFF: Deep sleep memory valid

Bits	Perm	Init	Description
7:0	RW	0	User defined data, reset to 0.

H Oscillator Configuration

The *Oscillator* is peripheral 4. The control registers are accessed using 8-bit reads and writes (use `write_periph_8(device, 4, ...)` and `read_periph_8(device, 4, ...)` for reads and writes).

Figure 47: Summary

Number	Perm	Description
0x00	RW	General oscillator control
0x01	RW	On-silicon-oscillator control
0x02	RW	Crystal-oscillator control

J.5 Power supply states whilst WAKING1: 0x10

This register controls what state the power control block should be in when in the WAKING1 state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state is entered if all enabled power supplies are good.

0x10: Power supply states whilst WAKING1

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	0	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

J.6 Power supply states whilst WAKING2: 0x14

This register controls what state the power control block should be in when in the WAKING2 state. It also defines the minimum time that the system shall stay in this state. When the minimum time is expired, the next state is entered if all enabled power supplies are good.

0x18: Power supply states whilst AWAKE

Bits	Perm	Init	Description
31:15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	1	Set to 1 to enable VOUT6 (IO supply).
4	RW	1	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	1	Set to 1 to enable DCDC1 (core supply).

J.8 Power supply states whilst SLEEPING1: 0x1C

This register controls what state the power control block should be in when in the SLEEPING1 state. It also defines the time that the system shall stay in this state.

Bits	Perm	Init	Description
31:21	RO	-	Reserved
20:16	RW	16	Log2 number of cycles to stay in this state: 0: 1 clock cycles 1: 2 clock cycles 2: 4 clock cycles ... 31: 2147483648 clock cycles
15	RO	-	Reserved
14	RW	0	Set to 1 to disable clock to the xCORE Tile.
13:10	RO	-	Reserved
9	RW	0	Sets modulation used by DCDC2: 0: PWM modulation (max 475 mA) 1: PFM modulation (max 50 mA)
8	RW	0	Sets modulation used by DCDC1: 0: PWM modulation (max 700 mA) 1: PFM modulation (max 50 mA)
7:6	RO	-	Reserved
5	RW	0	Set to 1 to enable VOUT6 (IO supply).
4	RW	0	Set to 1 to enable LDO5 (core PLL supply).
3:2	RO	-	Reserved
1	RO	1	Set to 1 to enable DCDC2 (analogue supply).
0	RW	0	Set to 1 to enable DCDC1 (core supply).

0x20: Power supply states whilst SLEEPING2

J.10 Power sequence status: 0x24

This register defines the current status of the power supply controller.

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:24	RW	2	Sets the power good level for VDDCORE and VDD1V8: 0: 0.80 x VDDCORE, 0.80 x VDD1V8 1: 0.85 x VDDCORE, 0.85 x VDD1V8 2: 0.90 x VDDCORE, 0.90 x VDD1V8 3: 0.75 x VDDCORE, 0.75 x VDD1V8
23:17	RO	-	Reserved
16	RW	0	Clear DCDC1 and DCDC2 error flags, not self clearing.
15	RO	-	Reserved
14:13	RW	0	Sets the DCDC2 current limit: 0: 1A 1: 1.5A 2: 2A 3: 0.5A
12:10	RO	-	Reserved
9:8	RW	1	Sets the clock used by DCDC2 to generate VDD1V8: 0: 0.9 MHz 1: 1.0 MHz 2: 1.1 MHz 3: 1.2 MHz
7	RO	-	Reserved
6:5	RW	0	Sets the DCDC1 current limit: 0: 1.2A 1: 1.8A 2: 2.5A 3: 0.8A
4:2	RO	-	Reserved
1:0	RW	1	Sets the clock used by DCDC1 to generate VDDCORE: 0: 0.9 MHz 1: 1.0 MHz 2: 1.1 MHz 3: 1.2 MHz

0x2C:
DCDC control

J.12 Power supply status: 0x30

This register provides the current status of the power supplies.

M 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 51 shows a decision diagram which explains what type of xSYS connectivity you need. The three subsections below explain the options in detail.

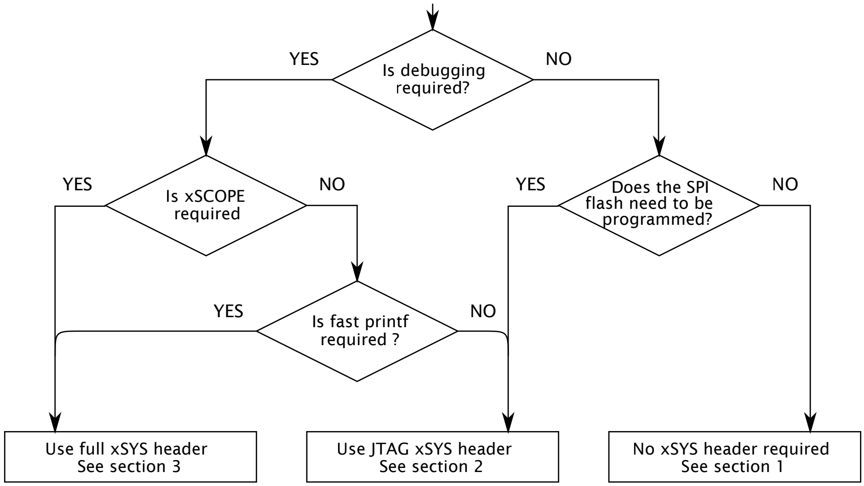

Figure 51: Decision diagram for the xSYS header

M.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.

M.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
- RST_N to pin 15 of the xSYS header
- If MODE2 is configured high, connect MODE2 to pin 3 of the xSYS header. Do not connect to VDDIO.
- If MODE3 is configured high, connect MODE3 to pin 3 of the xSYS header. Do not connect to VDDIO.

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

M.3 Full xSYS header

For a full xSYS header you will need to connect the pins as discussed in Section M.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 XLA, XLB, 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 XLB of tile 0 for xSCOPE I/O, you need to connect up XLB^1_{out}, XLB^0_{out}, XLB^0_{in}, XLB^1_{in} as follows:

- XLB^1_{out} (X0D16) to pin 6 of the xSYS header with a 33R series resistor close to the device.
- XLB^0_{out} (X0D17) to pin 10 of the xSYS header with a 33R series resistor close to the device.
- XLB^0_{in} (X0D18) to pin 14 of the xSYS header.
- XLB^1_{in} (X0D19) to pin 18 of the xSYS header.

O PCB Layout Design Check List

☑ This section is a checklist for use by PCB designers using the XS1-A6A-64-FB96. Each of the following sections contains items to check for each design.

O.1 Ground Balls and Ground Plane

☐ There is one via for each ground ball to minimize impedance and conduct heat away from the device (Section 15.1).

☐ There are only few non-ground vias around the square of ground balls, to creating a good, solid, ground plane.

O.2 Power supply decoupling

☐ VSUP has a ceramic X5R or X7R bulk decoupler as close as possible to the VSUP and PGND (VDDCORE) pins; right next to the device (Section 15).

☐ The 1V0 decoupling cap is close to the VDDCORE and PGND pins (Section 15).

☐ The 1V8 decoupling cap is close to the VDD1V8 and PGND pins (Section 15).

☐ All PGND nets are connected together prior to connection to the main ground plane (Section 15).

An example PCB layout is shown in Section 16. Placing the decouplers too far away may lead to the device not coming up, or not operating properly.