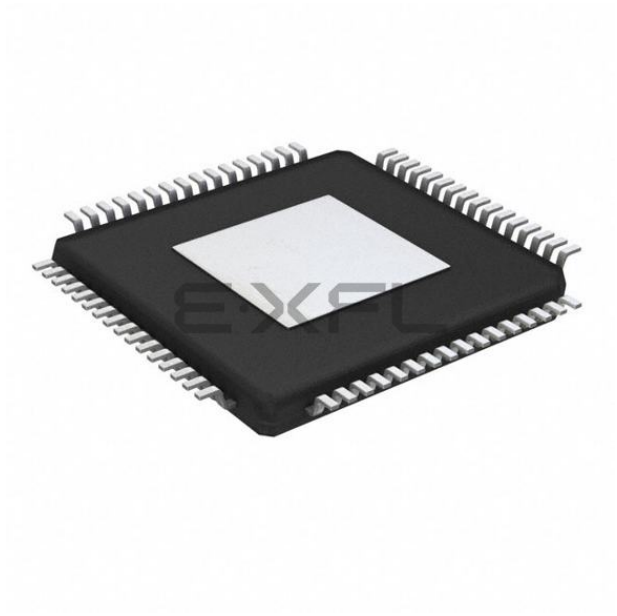

Welcome to E-XFL.COM

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 8-Core
Speed	1000MIPS
Connectivity	USB
Peripherals	-
Number of I/O	33
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	128K 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	64-TQFP Exposed Pad
Supplier Device Package	64-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xu208-128-tq64-c10

2 XU208-128-TQ64 Features

- **Multicore Microcontroller with Advanced Multi-Core RISC Architecture**
  - Eight real-time logical cores
  - Core share up to 500 MIPS
    - Up to 1000 MIPS in dual issue mode
  - Each logical core has:
    - Guaranteed throughput of between 1/5 and 1/8 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**
- **Programmable I/O**
  - 33 general-purpose I/O pins, configurable as input or output
    - Up to 9 x 1bit port, 5 x 4bit port, 3 x 8bit port, 1 x 16bit port
    - 1 xCONNECT link
  - 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**
  - 128KB internal single-cycle SRAM for code and data storage
  - 8KB internal OTP for application boot code
- **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**
  - 10: 500 MIPS
- **Power Consumption**
  - 170 mA (typical)
- **64-pin TQFP package 0.5 mm pitch**

4 Signal Description

This section lists the signals and I/O pins available on the XU208-128-TQ64. 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 has a weak pull-down or pull-up resistor. The resistor is enabled during and after reset. Enabling a link or port that uses the pin disables the resistor. Thereafter, the resistor can be enabled or disabled under software control. The resistor is designed to ensure defined logic input state for unconnected pins. It should not be used to pull external circuitry. Note that the resistors are highly non-linear and only a maximum pull current is specified in Section 13.2.
- ST: The IO pin has a Schmitt Trigger on its input.
- IOL/IOR: The IO pin is powered from VDDIOL, and VDDIOR respectively

Power pins (9)			
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_VDD	Digital tile power	PWR	
USB_VDD33	USB Analog power	PWR	
VDD	Digital tile power	PWR	
VDDIOL	Digital I/O power (left)	PWR	
VDDIOR	Digital I/O power (right)	PWR	

JTAG pins (5)			
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

(continued)

Signal	Function	Type	Properties

I/O pins (33)			
Signal	**Function**	**Type**	**Properties**
X0D00	$1A^0$	I/O	IOL, PD
X0D01	$1B^0$	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
X0D10	$1C^0$	I/O	IOL, PD
X0D11	$1D^0$	I/O	IOL, 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	$4D^0$ $8B^2$ $16A^{10}$	I/O	IOR, PD
X0D17	$4D^1$ $8B^3$ $16A^{11}$	I/O	IOR, PD
X0D18	$4D^2$ $8B^4$ $16A^{12}$	I/O	IOR, PD
X0D19	$4D^3$ $8B^5$ $16A^{13}$	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
X0D26	$4E^0$ $8C^0$ $16B^0$	I/O	IOR, PD
X0D27	$4E^1$ $8C^1$ $16B^1$	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
X0D35	$1L^0$	I/O	IOR, PD
X0D36	$1M^0$ $8D^0$ $16B^8$	I/O	IOL, PD
X0D37	$1N^0$ $8D^1$ $16B^9$	I/O	IOL, PD
X0D38	$1O^0$ $8D^2$ $16B^{10}$	I/O	IOL, PD
X0D39	$1P^0$ $8D^3$ $16B^{11}$	I/O	IOL, PD
X0D40	$X_0L0^1_{in}$ $8D^4$ $16B^{12}$	I/O	IOL, PD
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

usb pins (5)			
Signal	**Function**	**Type**	**Properties**
USB_DM	USB Serial Data Inverted	I/O	
USB_DP	USB Serial Data	I/O	
USB_ID	USB Device ID (OTG) - Reserved	I/O	

(continued)

Signal	Function	Type	Properties
USB_RTUNE	USB resistor	I/O	
USB_VBUS	USB Power Detect Pin	I/O	

System pins (1)			
Signal	**Function**	**Type**	**Properties**
CLK	PLL reference clock	Input	IOL, PD, ST

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

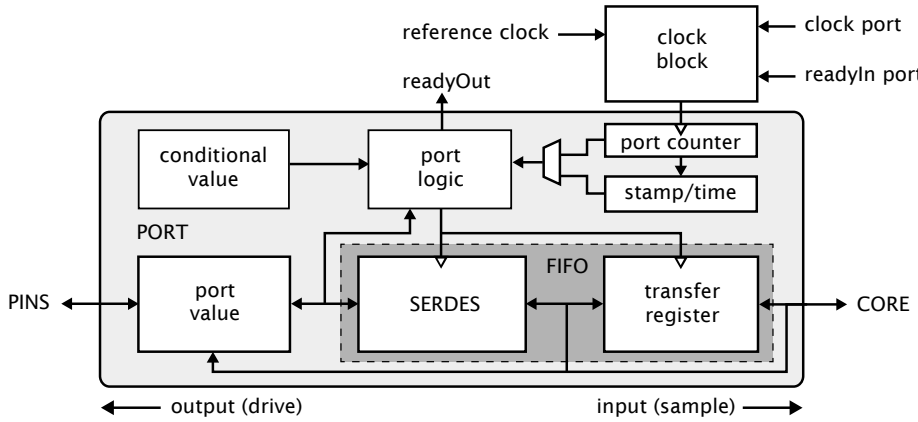

Figure 4: Port block diagram

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.

9 Memory

9.1 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.

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

The xCORE Tile integrates a single 128KB 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, XUD, 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 14. 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. 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.

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*.

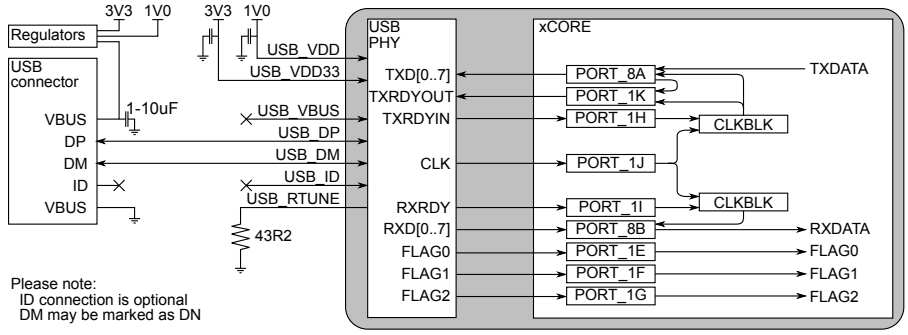

Figure 14: Bus powered *USB-device*

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.

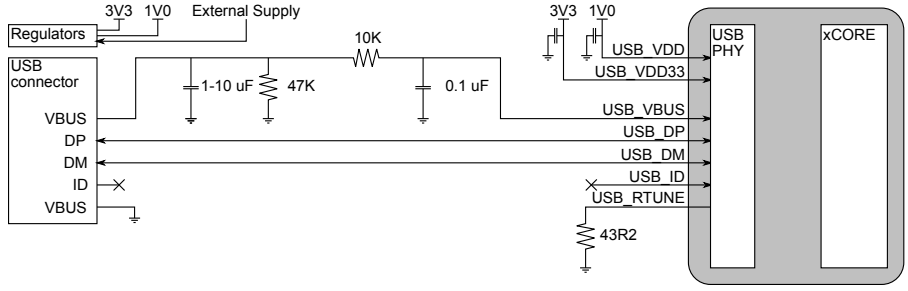

Figure 15: Self powered *USB-device*

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.

In any case, extra components (such as a ferrite bead and diodes) may be required for EMC compliance and ESD protection. Different wiring is required for USB-host and USB-OTG.

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 , *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				2				8				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, 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. VDDIOL must be a 3.3V 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:

- 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, 100nF 0402 for each supply pin). The ground side of the decoupling

Appendices

A Configuration of the XU208-128-TQ64

The device is configured through banks of registers, as shown in Figure 33.

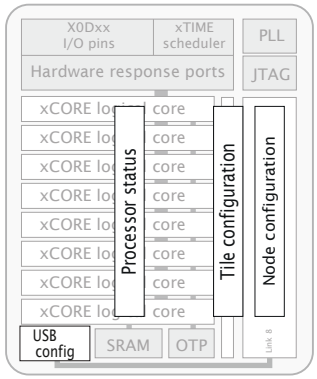

Figure 33: Registers

The following communication sequences specify how to access those registers. Any messages transmitted contain the most significant 24 bits of the channel-end to which a response is to be sent. This comprises the node-identifier and the channel number within the node. if no response is required on a write operation, supply 24-bits with the last 8-bits set, which suppresses the reply message. Any multi-byte data is sent most significant byte first.

A.1 Accessing a processor status register

The processor status registers are accessed directly from the processor instruction set. The instructions GETPS and SETPS read and write a word. The register number should be translated into a processor-status resource identifier by shifting the register number left 8 places, and ORing it with 0x0B. Alternatively, the functions `getps(reg)` and `setps(reg,value)` can be used from XC.

A.2 Accessing an xCORE Tile configuration register

xCORE Tile configuration registers can be accessed through the interconnect using the functions `write_tile_config_reg(tileref, ...)` and `read_tile_config_reg(tile ↪ ref, ...)`, where `tileref` is the name of the xCORE Tile, e.g. `tile[1]`. These functions implement the protocols described below.

Instead of using the functions above, a channel-end can be allocated to communicate with the xCORE tile configuration registers. The destination of the channel-end should be set to `0xnnnnC20C` where `nnnnnn` is the tile-identifier.

A write message comprises the following:

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).

0x02: xCORE Tile control

Bits	Perm	Init	Description
31:26	RO	-	Reserved
25:18	RW	0	RGMII TX data delay value (in PLL output cycle increments)
17:9	RW	0	RGMII TX clock divider value. TX clk rises when counter (clocked by PLL output) reaches this value and falls when counter reaches (value»1). Value programmed into this field should be actual divide value required minus 1
8	RW	0	Enable RGMII interface periph ports
7:6	RO	-	Reserved
5	RW	0	Select the dynamic mode (1) for the clock divider when the clock divider is enabled. In dynamic mode the clock divider is only activated when all active threads are paused. In static mode the clock divider is always enabled.
4	RW	0	Enable the clock divider. This divides the output of the PLL to facilitate one of the low power modes.
3	RO	-	Reserved
2	RW		Select between UTMI (1) and ULPI (0) mode.
1	RW		Enable the ULPI Hardware support module
0	RO	-	Reserved

B.4 xCORE Tile boot status: 0x03

This read-only register describes the boot status of the xCORE tile.

0x03: xCORE Tile boot status

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	RO		Processor number.
15:9	RO	-	Reserved
8	RO		Overwrite BOOT_MODE.
7:6	RO	-	Reserved
5	RO		Indicates if core1 has been powered off
4	RO		Cause the ROM to not poll the OTP for correct read levels
3	RO		Boot ROM boots from RAM
2	RO		Boot ROM boots from JTAG
1:0	RO		The boot PLL mode pin value.

0x07: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.8 Ring Oscillator Value: 0x08

This register contains the current count of the xCORE Tile Wire ring oscillator. This value is not reset on a system reset.

0x08: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.9 Ring Oscillator Value: 0x09

This register contains the current count of the Peripheral Cell ring oscillator. This value is not reset on a system reset.

0x09: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.10 Ring Oscillator Value: 0x0A

This register contains the current count of the Peripheral Wire ring oscillator. This value is not reset on a system reset.

0x0A: Ring Oscillator Value

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RO	0	Ring oscillator Counter data.

B.11 RAM size: 0x0C

The size of the RAM in bytes

0x0C: RAM size

Bits	Perm	Init	Description
31:2	RO		Most significant 16 bits of all addresses.
1:0	RO	-	Reserved

B.12 Debug SSR: 0x10

This register contains the value of the SSR register when the debugger was called.

0x10: Debug SSR

Bits	Perm	Init	Description
31:11	RO	-	Reserved
10	DRW		Address space indentifier
9	DRW		Determines the issue mode (DI bit) upon Kernel Entry after Exception or Interrupt.
8	RO		Determines the issue mode (DI bit).
7	DRW		When 1 the thread is in fast mode and will continually issue.
6	DRW		When 1 the thread is paused waiting for events, a lock or another resource.
5	RO	-	Reserved
4	DRW		1 when in kernel mode.
3	DRW		1 when in an interrupt handler.
2	DRW		1 when in an event enabling sequence.
1	DRW		When 1 interrupts are enabled for the thread.
0	DRW		When 1 events are enabled for the thread.

B.13 Debug SPC: 0x11

This register contains the value of the SPC register when the debugger was called.

0x11: Debug SPC

Bits	Perm	Init	Description
31:0	DRW		Value.

B.14 Debug SSP: 0x12

This register contains the value of the SSP register when the debugger was called.

B.25 Data breakpoint control register: 0x70 .. 0x73

This set of registers controls each of the four data watchpoints.

0x70 .. 0x73: Data breakpoint control register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:3	RO	-	Reserved
2	DRW	0	When 1 the breakpoints will be be triggered on loads.
1	DRW	0	Determines the break condition: 0 = A AND B, 1 = A OR B.
0	DRW	0	When 1 the instruction breakpoint is enabled.

B.26 Resources breakpoint mask: 0x80 .. 0x83

This set of registers contains the mask for the four resource watchpoints.

0x80 .. 0x83: Resources breakpoint mask

Bits	Perm	Init	Description
31:0	DRW		Value.

B.27 Resources breakpoint value: 0x90 .. 0x93

This set of registers contains the value for the four resource watchpoints.

0x90 .. 0x93: Resources breakpoint value

Bits	Perm	Init	Description
31:0	DRW		Value.

B.28 Resources breakpoint control register: 0x9C .. 0x9F

This set of registers controls each of the four resource watchpoints.

0x9C .. 0x9F: Resources breakpoint control register

Bits	Perm	Init	Description
31:24	RO	-	Reserved
23:16	DRW	0	A bit for each thread in the machine allowing the breakpoint to be enabled individually for each thread.
15:2	RO	-	Reserved
1	DRW	0	When 0 break when condition A is met. When 1 = break when condition B is met.
0	DRW	0	When 1 the instruction breakpoint is enabled.

C.24 SR of logical core 7: 0x67

Value of the SR of logical core 7

0x67: SR of logical core 7

Bits	Perm	Init	Description
31:0	CRO		Value.

0x06: PLL settings

Bits	Perm	Init	Description
31	RW		If set to 1, the chip will not be reset
30	RW		If set to 1, the chip will not wait for the PLL to re-lock. Only use this if a gradual change is made to the PLL
29	DW		If set to 1, set the PLL to be bypassed
28	DW		If set to 1, set the boot mode to boot from JTAG
27:26	RO	-	Reserved
25:23	RW		Output divider value range from 1 (8'h0) to 250 (8'hF9). P value.
22:21	RO	-	Reserved
20:8	RW		Feedback multiplication ratio, range from 1 (8'h0) to 255 (8'hFE). M value.
7	RO	-	Reserved
6:0	RW		Oscilator input divider value range from 1 (8'h0) to 32 (8'h0F). N value.

D.6 System switch clock divider: 0x07

Sets the ratio of the PLL clock and the switch clock.

0x07: System switch clock divider

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	0	SSwitch clock generation

D.7 Reference clock: 0x08

Sets the ratio of the PLL clock and the reference clock used by the node.

0x08: Reference clock

Bits	Perm	Init	Description
31:16	RO	-	Reserved
15:0	RW	3	Software ref. clock divider

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.
15:6	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.

0x40 .. 0x47: PLink status and network

D.17 Link configuration and initialization: 0x80 .. 0x88

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 0..7.

Bits	Perm	Init	Description
31	RW		Write to this bit with '1' will enable the XLink, writing '0' will disable it. This bit controls the muxing of ports with overlapping xlinks.
30	RW	0	0: operate in 2 wire mode; 1: operate in 5 wire mode
29:28	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	0	Specify min. number of idle system clocks between two continuous symbols witin a transmit token -1.
10:0	RW	0	Specify min. number of idle system clocks between two continuous transmit tokens -1.

0x80 .. 0x88: Link configuration and initialization

F USB PHY Configuration

The USB PHY is connected to the ports shown in section 10.

The *USB PHY* is peripheral 1. The control registers are accessed using 32-bit reads and writes (use `write_periph_32(device, 1, ...)` and `read_periph_32(device, 1, ...)` for reads and writes).

Number	Perm	Description
0x00	WO	UIFM reset
0x04	RW	UIFM IFM control
0x08	RW	UIFM Device Address
0x0C	RW	UIFM functional control
0x10	RW	UIFM on-the-go control
0x14	RO	UIFM on-the-go flags
0x18	RW	UIFM Serial Control
0x1C	RW	UIFM signal flags
0x20	RW	UIFM Sticky flags
0x24	RW	UIFM port masks
0x28	RW	UIFM SOF value
0x2C	RO	UIFM PID
0x30	RO	UIFM Endpoint
0x34	RW	UIFM Endpoint match
0x38	RW	OTG Flags mask
0x3C	RW	UIFM power signalling
0x40	RW	UIFM PHY control

Figure 38: Summary

F.1 UIFM reset: 0x00

A write to this register with any data resets all UIFM state, but does not otherwise affect the phy.

0x00: UIFM reset

Bits	Perm	Init	Description
31:0	WO		Value.

F.2 UIFM IFM control: 0x04

General settings of the UIFM IFM state machine.

H Schematics Design Check List

☑ This section is a checklist for use by schematics designers using the XU208-128-TQ64. 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 375 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 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.

☐ You have chosen an input clock frequency that is supported by the device (Section 7).