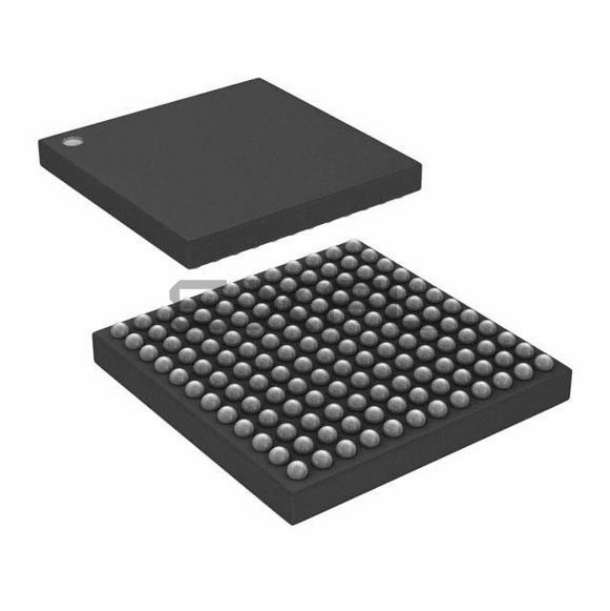
XMOS - XS1-G02B-FB144-C4 Datasheet



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 Dual-Core
Speed	400MIPS
Connectivity	Configurable
Peripherals	-
Number of I/O	88
Program Memory Size	128KB (32K x 32)
Program Memory Type	SRAM
EEPROM Size	-
RAM Size	-
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	144-LFBGA
Supplier Device Package	144-FBGA (11x11)
Purchase URL	https://www.e-xfl.com/product-detail/xmos/xs1-g02b-fb144-c4

1 Features

- Dual-Core Device with Advanced Multi-Threaded RISC Architecture
  - Up to 800 MIPS shared between up to 16 real-time threads
  - Each thread has:
    - Guaranteed throughput of between 1/4 and 1/8 of core MIPS
    - 16x32bit dedicated registers
  - 159 high-density 16/32-bit instructions
    - All have single clock-cycle execution (except for divide)
    - High-performance DSP (32x32 ! 64-bit MAC) and cryptographic instructions
- Programmable I/O
  - 88 general-purpose I/O pins, con gurable as input, output or bi-directional ports
  - Port sampling rates of up to 60 MHz with respect to an external clock
  - 64 channel ends for communication with other threads, on or o -chip
- Non-Volatile Memory
  - 128KB internal single-cycle SRAM (max 64KB per core) for code and data storage
  - 32KB internal OTP (max 8KB per core) for application boot code
- 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 ash memory
- Ambient Temperature Range
  - Commercial quali cation: 0 ℃ to 70 ℃
  - Industrial quali cation: -40 ℃ to 85 ℃
- Speed Grade
  - 400 MHz part: 400 MIPS
- 144-pin FBGA package 0.8 mm pitch

2 Pin Configuration

	1	2	3	4	5	6	7	8	9	10	11	12
A	IO VDD	X0D34	X0D35	X0D36	X0D37	X0D38	X0D39	X0D40	X0D41	X0D42	X0D43	VSS
B	X0D33	VSS	X0D18	X0D19	X0D20	IO VDD	VSS	X0D21	X0D22	X0D23	IO VDD	X2D24
C	X0D32	X0D17	VDD	X0D06	X0D07	X0D08	X0D09	X0D10	X0D11	VDD	X2D12	X2D25
D	X0D31	X0D16	X0D05	SS_PLL_BYPASS	SS_RESET	VDD	SS_OTP_VPP	SS_XC0_BS[0]	SS_XC_CFG[0]	X2D00	X2D13	X2D26
E	X0D30	X0D15	X0D04	SS_CLK	VSS	VSS	VSS	VSS	DEBUG_N	X2D01	X2D14	X2D27
F	X0D29	IO VDD	X0D03	SS_PLL_AGND	VSS	VSS	VSS	VSS	VDD	X2D02	VSS	X2D28
G	X0D28	VSS	X0D02	VDD	VSS	VSS	VSS	VSS	SS_TEST_ENA	X2D03	IO VDD	X2D29
H	X0D27	X0D14	X0D01	SS_PLL_AVDD	VSS	VSS	VSS	VSS	SS_TCK	X2D04	X2D15	X2D30
J	X0D26	X0D13	X0D00	SS_PLL_LOCK	SS_TMS	SS_TDO	VDD	SS_TDI	SS_TRST	X2D05	X2D16	X2D31
K	X0D25	X0D12	VDD	X2D11	X2D10	X2D09	X2D08	X2D07	X2D06	VDD	X2D17	X2D32
L	X0D24	IO VDD	X2D23	X2D22	X2D21	VSS	IO VDD	X2D20	X2D19	X2D18	VSS	X2D33
M	VSS	X2D43	X2D42	X2D41	X2D40	X2D39	X2D38	X2D37	X2D36	X2D35	X2D34	IO VDD

3 Signal Description

Module	Signal	Function	Type	Active	Properties
	PU=Pull Up, PD=Pull Down, ST=Schmitt Trigger, OT=Output Tristate, S=Switchable R_S=Required for SPI boot (§5.8), R_U=Required for USB-enabled devices (§10)				
Power	VSS	Digital ground	GND	—	
	VDD	Digital core power	PWR	—	
	IO VDD	Digital I/O power	PWR	—	
	SS_PLL_AGND	Analog ground for PLL	GND	—	
	SS_PLL_AVDD	Analog PLL power	PWR	—	
	SS_OTP_VPP	OTP programming voltage	PWR	—	
	SS_RESET	Global reset input	Input	—	PU, ST
PLL	SS_CLK	PLL reference clock	Input	—	PD, ST
	SS_PLL_BYPASS	PLL bypass	Input	—	PD
	SS_XC0_BS[0:0]	Boot status (core 0)	I/O	—	PU
JTAG	SS_TDI	Test data input	Input	—	PU, ST
	SS_TDO	Test data output	Output	—	PD
	SS_TMS	Test mode select	Input	—	PU, ST
	SS_TRST	Test reset input	Input	—	PU, ST
	SS_TCK	Test clock	Input	—	PU
	DEBUG_N	Multi-chip debug	I/O	Low	PU
XCore 0 I/O	X0D00	$P1A^{0}$	I/O	—	R_S
	X0D01	$X0LA^{4i}_{5b}$ $P1B^{0}$	I/O	—	R_S
	X0D02	$X0LA^{3i}_{5b}$ $P4A^{0}$ $P8A^{0}$ $P16A^{0}$ $P32A^{20}$	I/O	—	R_U
	X0D03	$X0LA^{2i}_{5b}$ $P4A^{1}$ $P8A^{1}$ $P16A^{1}$ $P32A^{21}$	I/O	—	R_U
	X0D04	$X0LA^{1i}_{2b/5b}$ $P4B^{0}$ $P8A^{2}$ $P16A^{2}$ $P32A^{22}$	I/O	—	R_U
	X0D05	$X0LA^{0i}_{2b/5b}$ $P4B^{1}$ $P8A^{3}$ $P16A^{3}$ $P32A^{23}$	I/O	—	R_U
	X0D06	$X0LA^{0o}_{2b/5b}$ $P4B^{2}$ $P8A^{4}$ $P16A^{4}$ $P32A^{24}$	I/O	—	R_U
	X0D07	$X0LA^{1o}_{2b/5b}$ $P4B^{3}$ $P8A^{5}$ $P16A^{5}$ $P32A^{25}$	I/O	—	R_U
	X0D08	$X0LA^{2o}_{5b}$ $P4A^{2}$ $P8A^{6}$ $P16A^{6}$ $P32A^{26}$	I/O	—	R_U
	X0D09	$X0LA^{3o}_{5b}$ $P4A^{3}$ $P8A^{7}$ $P16A^{7}$ $P32A^{27}$	I/O	—	R_U
	X0D10	$X0LA^{4o}_{5b}$ $P1C^{0}$	I/O	—	R_S
	X0D11	$P1D^{0}$	I/O	—	R_S
	X0D12	$P1E^{0}$	I/O	—	R_U
	X0D13	$X0LB^{4i}_{5b}$ $P1F^{0}$	I/O	—	R_U
	X0D14	$X0LB^{3i}_{5b}$ $P4C^{0}$ $P8B^{0}$ $P16A^{8}$ $P32A^{28}$	I/O	—	R_U
	X0D15	$X0LB^{2i}_{5b}$ $P4C^{1}$ $P8B^{1}$ $P16A^{9}$ $P32A^{29}$	I/O	—	R_U
	X0D16	$X0LB^{1i}_{2b/5b}$ $P4D^{0}$ $P8B^{2}$ $P16A^{10}$	I/O	—	R_U
	X0D17	$X0LB^{0i}_{2b/5b}$ $P4D^{1}$ $P8B^{3}$ $P16A^{11}$	I/O	—	R_U
	X0D18	$X0LB^{0o}_{2b/5b}$ $P4D^{2}$ $P8B^{4}$ $P16A^{12}$	I/O	—	R_U
	X0D19	$X0LB^{1o}_{2b/5b}$ $P4D^{3}$ $P8B^{5}$ $P16A^{13}$	I/O	—	R_U
	X0D20	$X0LB^{2o}_{5b}$ $P4C^{2}$ $P8B^{6}$ $P16A^{14}$ $P32A^{30}$	I/O	—	R_U
	X0D21	$X0LB^{3o}_{5b}$ $P4C^{3}$ $P8B^{7}$ $P16A^{15}$ $P32A^{31}$	I/O	—	R_U

(continued)

Module	Name	Function	Type	Active	Properties
XCore 0 I/O	X0D22	$X0LB^{4o}_{5b}$ $P1G^{0}$	I/O	—	R_U
	X0D23	$P1H^{0}$	I/O	—	R_U
	X0D24	$P1I^{0}$	I/O	—	
	X0D25	$P1J^{0}$	I/O	—	
	X0D26	$P4E^{0}$ $P8C^{0}$ $P16B^{0}$	I/O	—	R_U
	X0D27	$P4E^{1}$ $P8C^{1}$ $P16B^{1}$	I/O	—	R_U
	X0D28	$P4F^{0}$ $P8C^{2}$ $P16B^{2}$	I/O	—	R_U
	X0D29	$P4F^{1}$ $P8C^{3}$ $P16B^{3}$	I/O	—	R_U
	X0D30	$P4F^{2}$ $P8C^{4}$ $P16B^{4}$	I/O	—	R_U
	X0D31	$P4F^{3}$ $P8C^{5}$ $P16B^{5}$	I/O	—	R_U
	X0D32	$P4E^{2}$ $P8C^{6}$ $P16B^{6}$	I/O	—	R_U
	X0D33	$P4E^{3}$ $P8C^{7}$ $P16B^{7}$	I/O	—	R_U
	X0D34	$P1K^{0}$	I/O	—	
	X0D35	$P1L^{0}$	I/O	—	
	X0D36	$P1M^{0}$ $P8D^{0}$ $P16B^{8}$	I/O	—	
	X0D37	$P1N^{0}$ $P8D^{1}$ $P16B^{9}$	I/O	—	R_U
	X0D38	$P1O^{0}$ $P8D^{2}$ $P16B^{10}$	I/O	—	R_U
	X0D39	$P1P^{0}$ $P8D^{3}$ $P16B^{11}$	I/O	—	R_U
	X0D40	$P8D^{4}$ $P16B^{12}$	I/O	—	R_U
	X0D41	$P8D^{5}$ $P16B^{13}$	I/O	—	R_U
	X0D42	$P8D^{6}$ $P16B^{14}$	I/O	—	R_U
	X0D43	$P8D^{7}$ $P16B^{15}$	I/O	—	R_U
XCore 2 I/O	X2D00	$P1A^{0}$	I/O	—	
	X2D01	$X2LA^{4i}_{5b}$ $P1B^{0}$	I/O	—	
	X2D02	$X2LA^{3i}_{5b}$ $P4A^{0}$ $P8A^{0}$ $P16A^{0}$ $P32A^{20}$	I/O	—	R_U
	X2D03	$X2LA^{2i}_{5b}$ $P4A^{1}$ $P8A^{1}$ $P16A^{1}$ $P32A^{21}$	I/O	—	R_U
	X2D04	$X2LA^{1i}_{2b/5b}$ $P4B^{0}$ $P8A^{2}$ $P16A^{2}$ $P32A^{22}$	I/O	—	R_U
	X2D05	$X2LA^{0i}_{2b/5b}$ $P4B^{1}$ $P8A^{3}$ $P16A^{3}$ $P32A^{23}$	I/O	—	R_U
	X2D06	$X2LA^{0o}_{2b/5b}$ $P4B^{2}$ $P8A^{4}$ $P16A^{4}$ $P32A^{24}$	I/O	—	R_U
	X2D07	$X2LA^{1o}_{2b/5b}$ $P4B^{3}$ $P8A^{5}$ $P16A^{5}$ $P32A^{25}$	I/O	—	R_U
	X2D08	$X2LA^{2o}_{5b}$ $P4A^{2}$ $P8A^{6}$ $P16A^{6}$ $P32A^{26}$	I/O	—	R_U
	X2D09	$X2LA^{3o}_{5b}$ $P4A^{3}$ $P8A^{7}$ $P16A^{7}$ $P32A^{27}$	I/O	—	R_U
	X2D10	$X2LA^{4o}_{5b}$ $P1C^{0}$	I/O	—	
	X2D11	$P1D^{0}$	I/O	—	
	X2D12	$P1E^{0}$	I/O	—	R_U
	X2D13	$X2LB^{4i}_{5b}$ $P1F^{0}$	I/O	—	R_U
	X2D14	$X2LB^{3i}_{5b}$ $P4C^{0}$ $P8B^{0}$ $P16A^{8}$ $P32A^{28}$	I/O	—	R_U
	X2D15	$X2LB^{2i}_{5b}$ $P4C^{1}$ $P8B^{1}$ $P16A^{9}$ $P32A^{29}$	I/O	—	R_U
	X2D16	$X2LB^{1i}_{2b/5b}$ $P4D^{0}$ $P8B^{2}$ $P16A^{10}$	I/O	—	R_U
	X2D17	$X2LB^{0i}_{2b/5b}$ $P4D^{1}$ $P8B^{3}$ $P16A^{11}$	I/O	—	R_U
	X2D18	$X2LB^{0o}_{2b/5b}$ $P4D^{2}$ $P8B^{4}$ $P16A^{12}$	I/O	—	R_U
	X2D19	$X2LB^{1o}_{2b/5b}$ $P4D^{3}$ $P8B^{5}$ $P16A^{13}$	I/O	—	R_U
	X2D20	$X2LB^{2o}_{5b}$ $P4C^{2}$ $P8B^{6}$ $P16A^{14}$ $P32A^{30}$	I/O	—	R_U

(continued)

Module	Name	Function	Type	Active	Properties
XCore 2 I/O	X2D21	$X2LB_{5b}^{3o}$ $P4C^{3}$ $P8B^{7}$ $P16A^{15}$ $P32A^{31}$	I/O	—	R_U
	X2D22	$X2LB_{5b}^{4o}$ $P1G^{0}$	I/O	—	R_U
	X2D23	$P1H^{0}$	I/O	—	R_U
	X2D24	$P1I^{0}$	I/O	—	
	X2D25	$P1J^{0}$	I/O	—	
	X2D26	$P4E^{0}$ $P8C^{0}$ $P16B^{0}$	I/O	—	R_U
	X2D27	$P4E^{1}$ $P8C^{1}$ $P16B^{1}$	I/O	—	R_U
	X2D28	$P4F^{0}$ $P8C^{2}$ $P16B^{2}$	I/O	—	R_U
	X2D29	$P4F^{1}$ $P8C^{3}$ $P16B^{3}$	I/O	—	R_U
	X2D30	$P4F^{2}$ $P8C^{4}$ $P16B^{4}$	I/O	—	R_U
	X2D31	$P4F^{3}$ $P8C^{5}$ $P16B^{5}$	I/O	—	R_U
	X2D32	$P4E^{2}$ $P8C^{6}$ $P16B^{6}$	I/O	—	R_U
	X2D33	$P4E^{3}$ $P8C^{7}$ $P16B^{7}$	I/O	—	R_U
	X2D34	$P1K^{0}$	I/O	—	
	X2D35	$P1L^{0}$	I/O	—	
	X2D36	$P1M^{0}$ $P8D^{0}$ $P16B^{8}$	I/O	—	
	X2D37	$P1N^{0}$ $P8D^{1}$ $P16B^{9}$	I/O	—	R_U
	X2D38	$P1O^{0}$ $P8D^{2}$ $P16B^{10}$	I/O	—	R_U
	X2D39	$P1P^{0}$ $P8D^{3}$ $P16B^{11}$	I/O	—	R_U
	X2D40	$P8D^{4}$ $P16B^{12}$	I/O	—	R_U
	X2D41	$P8D^{5}$ $P16B^{13}$	I/O	—	R_U
	X2D42	$P8D^{6}$ $P16B^{14}$	I/O	—	R_U
	X2D43	$P8D^{7}$ $P16B^{15}$	I/O	—	R_U
Reserved	SS_PLL_LOCK	Reserved (do not connect)	Output	—	PD
	SS_TEST_ENA	Reserved (tie to VSS)	Input	—	PD
	SS_XC_CFG[0:0]	Reserved (tie to IO VDD)	Input	—	PD

4 Block Diagram

5 Product Overview

The XMOS XS1-G02B-FB144 is a powerful device that provides a simple design process and highly-flexible solution to many applications. The device consists of two XCores, each comprising an event-driven processor with tightly integrated I/O and on-chip memory. The processors run mutiple tasks simultaneously using hardware threads, each of which is guaranteed a slice of processing power and can execute computational code, control software and I/O interfaces. Threads use channels to exchange data within a core or across cores. The cores are connected via an integrated switch network, which uses a proprietary physical layer protocol, and which can also be used to add additional resources to a design. The I/O pins are driven using intelligent ports that can serialize data, interpret strobe signals and wait for scheduled times or events, making the device ideal for real-time control applications.

The device can be configured using a set of software components that are rapidly customized and composed. XMOS provides source code libraries for many standard components. The device can be programmed using high-level languages such as C/C++ and the XMOS-originated XC language. XC provides extensions to C that simplify the control over concurrency, I/O and time.

The XMOS toolchain includes compilers, a simulator, debugger and static timing analyzer. The combination of real-time software, a compiler and timing analyzer enables the programmer to close timings on components of the design without a detailed understanding of the hardware characteristics.

5.1 Threads, Synchronizers and Locks

Each XCore has up to eight active threads, which issue instructions down a shared four-stage pipeline. Instructions from the active threads are issued round-robin. If up to four threads are active, each thread is allocated a quarter of the processing cycles. If more than four threads are active, each thread is allocated at least $^1/_n$ cycles (for *n* threads). Figure 1 shows the guaranteed thread performance depending on the number of threads used.

Figure 1: Thread performance

Speed Grade	Minimum MIPS per thread (for ***n*** threads)							
	1	2	3	4	5	6	7	8
400 MHz	100	100	100	100	80	67	57	50

There is no way that the performance of a thread can be reduced below these predicted levels. Because threads may be delayed on I/O, however, their unused processor cycles can be taken by other threads. This means that for more than four threads, the performance of each thread is often higher than the predicted minimum.

5.2 Channel Ends, Links and Switch

Threads communicate using point-to-point connections formed between two channel ends. Between cores, channel communications are implemented over XMOS Links and routed through switches. The links operate in either 2bit/direction or 5bit/direction mode, depending on the amount of bandwidth required. Circuit switched, streaming and packet switched data can both be supported e□ciently. Streams provide the fastest possible data rates between XCores (up to 250 MBit/s), but each stream requires a single link to be reserved between switches on two cores. All packet communications can be multiplexed onto a single link. A total of eight 5bit links are available between both cores.

Information on the supported routing topologies that can be used to connect multiple devices together can be found in the XS1-G Link Performance and Design Guides, document number X2215.

5.3 Ports and Clock Blocks

Ports provide an interface between the threads and I/O pins. The operation of each port is synchronized to a clock block. A clock block can be connected to an external clock input, or it can be run from the divided reference clock. A clock block can also output its signal to a pin. On reset, each port is connected to clock block 0, which runs from the reference clock.

The ports and links are multiplexed, allowing the pins to be configured for use by ports of di□erent widths or links. If an XMOS 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.

5.4 Timers

Timers are 32-bit counters that are relative to the reference clock. A timer is defined to tick every 10 ns. This value is derived from the reference clock, which is configured to tick at 100 MHz by default.

5.5 SRAM

Each XCore integrates a single 64 KB 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 core clock cycle. There is no dedicated external memory interface, although data memory can be expanded through appropriate use of the ports.

5.6 OTP

Each XCore integrates 8 KB one-time programmable (OTP) memory along with a security register that configures system wide security features. The OTP holds data in 2k rows x 32-bit configuration 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.

5.6.1 Security Register

The security register enables the following security features:

Secure Boot: The XCore is forced to boot from address 0 of the OTP, allowing the XCore boot ROM to be bypassed (*see* §5.8). This feature can be used to implement a secure bootloader which loads an encrypted image from external flash, decrypts and CRC checks it with the processor, and discontinues the boot process if the decryption or CRC check fails. XMOS provides a default secure bootloader that can be written to the OTP along with secret decryption keys.

Disable JTAG: The JTAG interface is disabled, making it impossible for the processor state or memory content to be accessed via the JTAG interface.

Disable Link access: Other processors are forbidden access to the processor state via the system switch.

Disabling both JTAG and Link access transforms a core into a “secure island” with other cores free for non-secure user application code.

Disable Global Debug access: Disables access to the SS_DEBUG pin.

OTP Master and Sector Lock: Further access to the OTP is prevented by setting the master lock. Locks can also be applied to each of the four OTP sectors individually.

These security features provide a strong level of protection and are su□cient for providing strong IP security.

5.7 PLL

The PLL is used to generate all on-chip clocks. SS_CLK is the reference clock input. It should be supplied with a clock with monotonic rising edges and should be stable before SS_RESET is taken high.

Many standard clock frequencies can be used with appropriate settings configured into the PLL. At boot time, before the PLL can be reconfigured, the PLL multiplier is set using the pins specified in the table in Figure 2. The PLL increases the clock frequency to the core frequency used to run the processor data path and the switch.