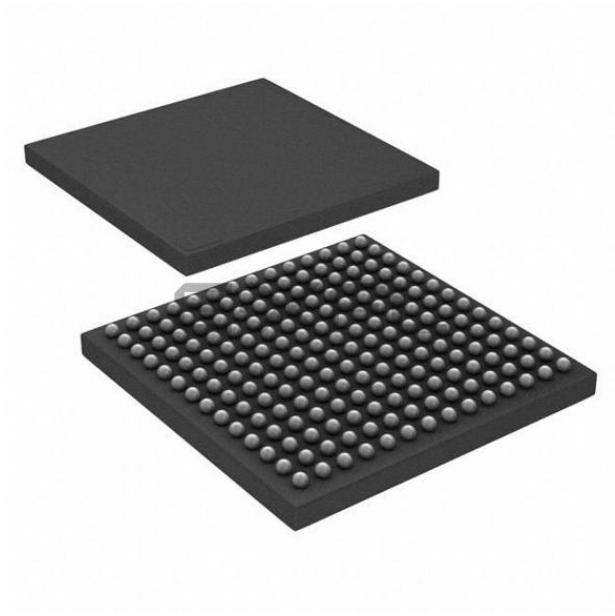
Microchip Technology - AT32AP7002-CTUR Datasheet



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	AVR
Core Size	32-Bit Single-Core
Speed	150MHz
Connectivity	EBI/EMI, I²C, Memory Card, PS/2, SPI, SSC, UART/USART, USB
Peripherals	AC'97, DMA, I²S, LCD, POR, PWM, WDT
Number of I/O	85
Program Memory Size	-
Program Memory Type	ROMless
EEPROM Size	-
RAM Size	32K x 8
Voltage - Supply (Vcc/Vdd)	1.65V ~ 1.95V
Data Converters	D/A 2x16b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	196-TFBGA, CSBGA
Supplier Device Package	196-CTBGA (12x12)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/at32ap7002-ctur

1. Part Description

The AT32AP7002 is a complete System-on-chip application processor with an AVR32 RISC processor achieving 210 DMIPS running 150 MHz. AVR32 is a high-performance 32-bit RISC microprocessor core, designed for cost-sensitive embedded applications, with particular emphasis on low power consumption, high code density and high application performance.

AT32AP7002 implements a Memory Management Unit (MMU) and a flexible interrupt controller supporting modern operating systems and real-time operating systems. The processor also includes a rich set of DSP and SIMD instructions, specially designed for multimedia and telecom applications.

AT32AP7002 incorporates SRAM memories on-chip for fast and secure access. For applications requiring additional memory, external 16-bit SRAM is accessible. Additionally, an SDRAM controller provides off-chip volatile memory access as well as controllers for all industry standard off-chip non-volatile memories, like Compact Flash, Multi Media Card (MMC), Secure Digital (SD)-card, SmartCard, NAND Flash and Atmel DataFlash™.

The Direct Memory Access controller for all the serial peripherals enables data transfer between memories without processor intervention. This reduces the processor overhead when transferring continuous and large data streams between modules in the MCU.

The Timer/Counters includes three identical 16-bit timer/counter channels. Each channel can be independently programmed to perform a wide range of functions including frequency measurement, event counting, interval measurement, pulse generation, delay timing and pulse width modulation.

AT32AP7002 also features an onboard LCD Controller, supporting single and double scan monochrome and color passive STN LCD modules and single scan active TFT LCD modules. On monochrome STN displays, up to 16 gray shades are supported using a time-based dithering algorithm and Frame Rate Control (FRC) method. This method is also used in color STN displays to generate up to 4096 colors.

The LCD Controller is programmable for supporting resolutions up to 2048 x 2048 with a pixel depth from 1 to 24 bits per pixel.

A pixel co-processor provides color space conversions for images and video, in addition to a wide variety of hardware filter support

The media-independent interface (MII) and reduced MII (RMII) 10/100 Ethernet MAC modules provides on-chip solutions for network-connected devices.

Synchronous Serial Controllers provide easy access to serial communication protocols, audio standards like I2S and frame-based protocols.

The Java hardware acceleration implementation in AVR32 allows for a very high-speed Java byte-code execution. AVR32 implements Java instructions in hardware, reusing the existing RISC data path, which allows for a near-zero hardware overhead and cost with a very high performance.

The Image Sensor Interface supports cameras with up to 12-bit data buses.

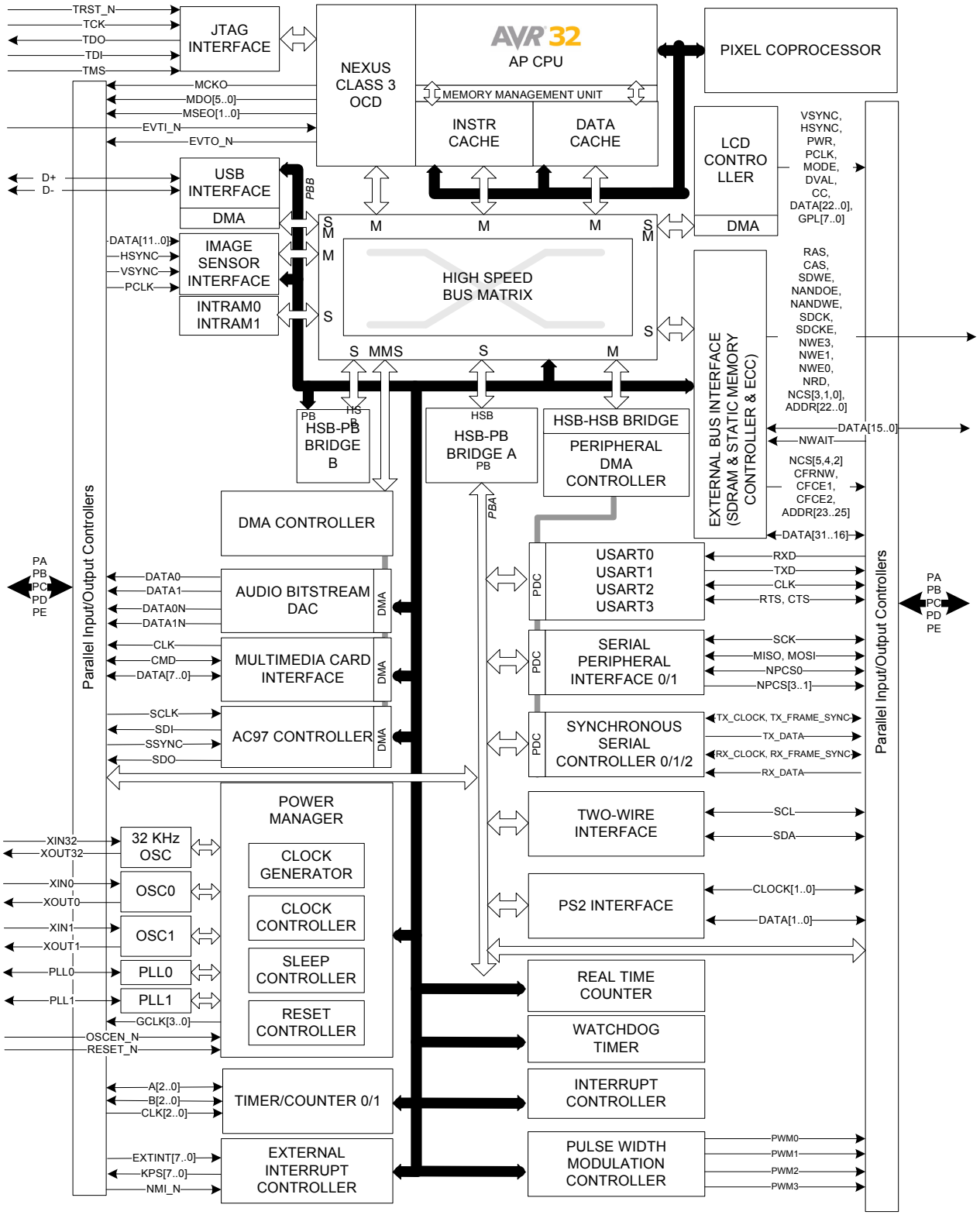
PS2 connectivity is provided for standard input devices like mice and keyboards.

AT32AP7002 integrates a class 3 Nexus 2.0 On-Chip Debug (OCD) System, with non-intrusive real-time trace, full-speed read/write memory access in addition to basic runtime control.

The C-compiler is closely linked to the architecture and is able to utilize code optimization features, both for size and speed.

2. Blockdiagram

Figure 2-1. Blockdiagram



2.0.1 AVR32AP CPU

- 32-bit load/store AVR32B RISC architecture.
  - Up to 15 general-purpose 32-bit registers.
  - 32-bit Stack Pointer, Program Counter and Link Register reside in register file.
  - Fully orthogonal instruction set.
  - Privileged and unprivileged modes enabling efficient and secure Operating Systems.
  - Innovative instruction set together with variable instruction length ensuring industry leading code density.
  - DSP extention with saturating arithmetic, and a wide variety of multiply instructions.
  - SIMD extention for media applications.
- 7 stage pipeline allows one instruction per clock cycle for most instructions.
  - Java Hardware Acceleration.
  - Byte, half-word, word and double word memory access.
  - Unaligned memory access.
  - Shadowed interrupt context for INT3 and multiple interrupt priority levels.
  - Dynamic branch prediction and return address stack for fast change-of-flow.
  - Coprocessor interface.
- Full MMU allows for operating systems with memory protection.
- 16Kbyte Instruction and 16Kbyte data caches.
  - Virtually indexed, physically tagged.
  - 4-way associative.
  - Write-through or write-back.
- Nexus Class 3 On-Chip Debug system.
  - Low-cost NanoTrace supported.

2.0.2 Pixel Coprocessor (PICO)

- Coprocessor coupled to the AVR32 CPU Core through the TCB Bus.
  - Coprocessor number one on the TCB bus.
- Three parallel Vector Multiplication Units (VMU) where each unit can:
  - Multiply three pixel components with three coefficients.
  - Add the products from the multiplications together.
  - Accumulate the result or add an offset to the sum of the products.
- Can be used for accelerating:
  - Image Color Space Conversion.
    - Configurable Conversion Coefficients.
    - Supports packed and planar input and output formats.
    - Supports subsampled input color spaces (i.e 4:2:2, 4:2:0).
  - Image filtering/scaling.
    - Configurable Filter Coefficients.
    - Throughput of one sample per cycle for a 9-tap FIR filter.
    - Can use the built-in accumulator to extend the FIR filter to more than 9-taps.
    - Can be used for bilinear/bicubic interpolations.
  - MPEG-4/H.264 Quarter Pixel Motion Compensation.
- Flexible input Pixel Selector.
  - Can operate on numerous different image storage formats.
- Flexible Output Pixel Inserter.
  - Scales and saturates the results back to 8-bit pixel values.
  - Supports packed and planar output formats.

- Configurable coefficients with flexible fixed-point representation.

2.0.3 Debug and Test system

- IEEE1149.1 compliant JTAG and boundary scan
- Direct memory access and programming capabilities through JTAG interface
- Extensive On-Chip Debug features in compliance with IEEE-ISTO 5001-2003 (Nexus 2.0) Class 3
- Auxiliary port for high-speed trace information
- Hardware support for 6 Program and 2 data breakpoints
- Unlimited number of software breakpoints supported
- Advanced Program, Data, Ownership, and Watchpoint trace supported

2.0.4 DMA Controller

- 2 HSB Master Interfaces
- 3 Channels
- Software and Hardware Handshaking Interfaces
  - 11 Hardware Handshaking Interfaces
- Memory/Non-Memory Peripherals to Memory/Non-Memory Peripherals Transfer
- Single-block DMA Transfer
- Multi-block DMA Transfer
  - Linked Lists
  - Auto-Reloading
  - Contiguous Blocks
- DMA Controller is Always the Flow Controller
- Additional Features
  - Scatter and Gather Operations
  - Channel Locking
  - Bus Locking
  - FIFO Mode
  - Pseudo Fly-by Operation

2.0.5 Peripheral DMA Controller

- Transfers from/to peripheral to/from any memory space without intervention of the processor.
- Next Pointer Support, forbids strong real-time constraints on buffer management.
- Eighteen channels
  - Two for each USART
  - Two for each Serial Synchronous Controller
  - Two for each Serial Peripheral Interface

2.0.6 Bus system

- HSB bus matrix with 10 Masters and 8 Slaves handled
  - Handles Requests from the CPU Icache, CPU Dcache, HSB bridge, HISI, USB 2.0 Controller, LCD Controller, DMA Controller 0, DMA Controller 1, and to internal SRAM 0, internal SRAM 1, PB A, PB B, EBI and, USB.

Table 2-2. CTBGA196 Package Pinout A9..T16

	9	10	11	12	13	14
A	PB18	PB16	PB11	PB08	PB05	PB04
B	PB21	PB17	PB12	PB09	PB06	PB03
C	PA06	PB19	PB14	PB10	PB07	PB02
D	VDDIO	PB20	PB15	PB13	GND	PB01
E	PA07	GND	PB00	PX44	VDDIO	PX45
F	PX42	GND	GND	PX43	PX46	PX40
G	PX30	PX25	PX31	VDDIO	VDDCORE	PX39
H	PA28	PX20	PX28	PX29	VDDCORE	PX26
J	PB27	PX37	PX23	PX27	PX21	PX24
K	PB28	PX15	PX36	PX19	PX34	PX18
L	PX41	GND	PX07	VDDIO	GND	PX35
M	PX53	PX06	PX11	PX12	PX17	PX14
N	PB26	VDDIO	PX09	PB29	PX16	PX13
P	PA27	PA31	PX52	PX08	PX10	PB30

Table 3-1. Signal Description List

Signal Name	Function	Type	Active Level	Comments
DATA0 - DATA7	Multimedia Card Data	I/O		
Parallel Input/Output - PIOA, PIOB, PIOC, PIOD				
PA0 - PA31	Parallel I/O Controller PIOA	I/O		
PB0 - PB30	Parallel I/O Controller PIOB	I/O		
PC20 - PC23/ PC28 - PC31	Parallel I/O Controller PIOC	I/O		
PD0 - PD17	Parallel I/O Controller PIOD	I/O		
PS2 Interface - PSIF				
CLOCK0 - CLOCK1	PS2 Clock	Input		
DATA0 - DATA1	PS2 Data	I/O		
Serial Peripheral Interface - SPI0, SPI1				
MISO	Master In Slave Out	I/O		
MOSI	Master Out Slave In	I/O		
NPCS0 - NPCS3	SPI Peripheral Chip Select	I/O	Low	
SCK	Clock	Output		
Synchronous Serial Controller - SSC0, SSC1, SSC2				
RX_CLOCK	SSC Receive Clock	I/O		
RX_DATA	SSC Receive Data	Input		
RX_FRAME_SYNC	SSC Receive Frame Sync	I/O		
TX_CLOCK	SSC Transmit Clock	I/O		
TX_DATA	SSC Transmit Data	Output		
TX_FRAME_SYNC	SSC Transmit Frame Sync	I/O		
DMA Controller - DMACA				
DMARQ0 - DMARQ3	DMA Requests	Input		
Timer/Counter - TIMER0, TIMER1				
A0	Channel 0 Line A	I/O		
A1	Channel 1 Line A	I/O		
A2	Channel 2 Line A	I/O		
B0	Channel 0 Line B	I/O		

7. Peripherals

7.1 Peripheral address map

Table 7-1. Peripheral Address Mapping

Address		Peripheral Name	Bus
0xFF000000	LCDC	LCD Controller Slave Interface - LCDC	HSB
0xFF200000	DMACA	DMA Controller Slave Interface- DMACA	HSB
0xFF300000	USBA	USB Slave Interface - USBA	HSB
0xFFE00000	SPI0	Serial Peripheral Interface - SPI0	PB A
0xFFE00400	SPI1	Serial Peripheral Interface - SPI1	PB A
0xFFE00800	TWI	Two-wire Interface - TWI	PB A
0xFFE00C00	USART0	Universal Synchronous Asynchronous Receiver Transmitter - USART0	PB A
0xFFE01000	USART1	Universal Synchronous Asynchronous Receiver Transmitter - USART1	PB A
0xFFE01400	USART2	Universal Synchronous Asynchronous Receiver Transmitter - USART2	PB A
0xFFE01800	USART3	Universal Synchronous Asynchronous Receiver Transmitter - USART3	PB A
0xFFE01C00	SSC0	Synchronous Serial Controller - SSC0	PB A
0xFFE02000	SSC1	Synchronous Serial Controller - SSC1	PB A
0xFFE02400	SSC2	Synchronous Serial Controller - SSC2	PB A
0xFFE02800	PIOA	Parallel Input/Output 2 - PIOA	PB A
0xFFE02C00	PIOB	Parallel Input/Output 2 - PIOB	PB A
0xFFE03000	PIOC	Parallel Input/Output 2 - PIOC	PB A
0xFFE03400	PIOD	Parallel Input/Output 2 - PIOD	PB A

Table 7-2. Interrupt Request Signal Map

Group	Line	Signal
19	0	EIC0
	1	EIC1
	2	EIC2
	3	EIC3
20	0	PM
21	0	RTC
22	0	TC00
	1	TC01
	2	TC02
23	0	TC10
	1	TC11
	2	TC12
24	0	PWM
27	0	ABDAC
28	0	MCI
29	0	AC97C
30	0	ISI
31	0	USBA
32	0	EBI

7.3 DMACA Handshake Interface Map

The following table details the hardware handshake map between the DMACA and the peripherals attached to it: :

Table 7-3. Hardware Handshaking Connection

Request	Hardware Handshaking Interface
MCI RX	0
MCI TX	1
ABDAC TX	2
AC97C CHANNEL A RX	3
AC97C CHANNEL A TX	4
AC97C CHANNEL B RX	5
AC97C CHANNEL B TX	6
EXTERNAL DMA REQUEST 0	7
EXTERNAL DMA REQUEST 1	8
EXTERNAL DMA REQUEST 2	9
EXTERNAL DMA REQUEST 3	10

7.4.3 SPIs

Each SPI can be connected to an internally divided clock:

Table 7-6. SPI clock connections

SPI	Source	Name	Connection
0	Internal	CLK_DIV	clk_pba / 32
1			

7.4.4 USBA

OSC1 is connected to the USB HS Phy and must be 12 MHz when using the USBA.

7.5 External Interrupt Pin Mapping

External interrupt requests are connected to the following pins::

Table 7-7. External Interrupt Pin Mapping

Source	Connection
NMI_N	PB24
EXTINT0	PB25
EXTINT1	PB26
EXTINT2	PB27
EXTINT3	PB28

7.6 Nexus OCD AUX port connections

If the OCD trace system is enabled, the trace system will take control over a number of pins, irrespectively of the PIO configuration. Two different OCD trace pin mappings are possible, depending on the configuration of the OCD AXS register. For details, see the *AVR32 AP Technical Reference Manual.*

Table 7-8. Nexus OCD AUX port connections

Pin	AXS=0	AXS=1
EVTI_N	EVTI_N	EVTI_N
MDO[5]	PB09	PC18
MDO[4]	PB08	PC14
MDO[3]	PB07	PC12
MDO[2]	PB06	PC11
MDO[1]	PB05	PC06
MDO[0]	PB04	PC05
EVTO_N	PB03	PB28
MCKO	PB02	PC02
MSEO[1]	PB01	PC01
MSEO[0]	PB00	PC00

7.7 Peripheral Multiplexing on IO lines

The AT32AP7002 features five PIO controllers, PIOA to PIOE, that multiplex the I/O lines of the peripheral set. Each PIO Controller controls up to thirty-two lines.

Each line can be assigned to one of two peripheral functions, A or B. The tables in the following pages define how the I/O lines of the peripherals A and B are multiplexed on the PIO Controllers.

Note that some output only peripheral functions might be duplicated within the tables.

7.7.1 PIO Controller A Multiplexing

Table 7-9. PIO Controller A Multiplexing

CTBGA196	I/O Line	Peripheral A	Peripheral B
J3	PA00	SPI0 - MISO	SSC1 - RX_FRAME_SYNC
J1	PA01	SPI0 - MOSI	SSC1 - TX_FRAME_SYNC
G6	PA02	SPI0 - SCK	SSC1 - TX_CLOCK
J2	PA03	SPI0 - NPCS[0]	SSC1 - RX_CLOCK
G5	PA04	SPI0 - NPCS[1]	SSC1 - TX_DATA
K1	PA05	SPI0 - NPCS[2]	SSC1 - RX_DATA
C9	PA06	TWI - SDA	USART0 - RTS
E9	PA07	TWI - SCL	USART0 - CTS
G7	PA08	PSIF - CLOCK	USART0 - RXD
J6	PA09	PSIF - DATA	USART0 - TXD
H6	PA10	MCI - CLK	USART0 - CLK
K2	PA11	MCI - CMD	TC0 - CLK0
K3	PA12	MCI - DATA[0]	TC0 - A0
M1	PA13	MCI - DATA[1]	TC0 - A1
H7	PA14	MCI - DATA[2]	TC0 - A2
N1	PA15	MCI - DATA[3]	TC0 - B0
K4	PA16	USART1 - CLK	TC0 - B1
P1	PA17	USART1 - RXD	TC0 - B2
J7	PA18	USART1 - TXD	TC0 - CLK2
L3	PA19	USART1 - RTS	TC0 - CLK1
N2	PA20	USART1 - CTS	SPI0 - NPCS[3]
L2	PA21	SSC0 - RX_FRAME_SYNC	PWM - PWM[2]
M2	PA22	SSC0 - RX_CLOCK	PWM - PWM[3]
M3	PA23	SSC0 - TX_CLOCK	TC1 - A0
P2	PA24	SSC0 - TX_FRAME_SYNC	TC1 - A1
L7	PA25	SSC0 - TX_DATA	TC1 - B0
K7	PA26	SSC0 - RX_DATA	TC1 - B1
P9	PA27	SPI1 - NPCS[3]	TC1 - CLK0
H9	PA28	PWM - PWM[0]	TC1 - A2

7.7.3 PIO Controller C Multiplexing

Table 7-11. PIO Controller C Multiplexing

CTBGA196	I/O Line	Peripheral A	Peripheral B
A7	PC20	LCDC - HSYNC	
C8	PC21	LCDC - PCLK	
B7	PC22	LCDC - VSYNC	
A6	PC23	LCDC - DVAL	
A5	PC28	LCDC - DATA[2]	
C7	PC29	LCDC - DATA[3]	
D8	PC30	LCDC - DATA[4]	
E8	PC31	LCDC - DATA[5]	

7.7.4 PIO Controller D Multiplexing

Table 7-12. PIO Controller D Multiplexing

CTBGA196	I/O Line	Peripheral A	Peripheral B
D2	PD00	LCDC - DATA[6]	
C2	PD01	LCDC - DATA[7]	
F4	PD04	LCDC - DATA[10]	
G1	PD05	LCDC - DATA[11]	
G3	PD06	LCDC - DATA[12]	
E5	PD07	LCDC - DATA[13]	
G2	PD08	LCDC - DATA[14]	
F5	PD09	LCDC - DATA[15]	
N3	PD12	LCDC - DATA[18]	
P3	PD13	LCDC - DATA[19]	
P4	PD14	LCDC - DATA[20]	
N4	PD15	LCDC - DATA[21]	
N5	PD16	LCDC - DATA[22]	
M4	PD17	LCDC - DATA[23]	

7.8.7 USART

- Programmable Baud Rate Generator
- 5- to 9-bit full-duplex synchronous or asynchronous serial communications
  - 1, 1.5 or 2 stop bits in Asynchronous Mode or 1 or 2 stop bits in Synchronous Mode
  - Parity generation and error detection
  - Framing error detection, overrun error detection
  - MSB- or LSB-first
  - Optional break generation and detection
  - By 8 or by-16 over-sampling receiver frequency
  - Hardware handshaking RTS-CTS
  - Receiver time-out and transmitter timeguard
  - Optional Multi-drop Mode with address generation and detection
  - Optional Manchester Encoding
- RS485 with driver control signal
- ISO7816, T = 0 or T = 1 Protocols for interfacing with smart cards
  - NACK handling, error counter with repetition and iteration limit
- IrDA modulation and demodulation
  - Communication at up to 115.2 Kbps
- Test Modes 46
  - Remote Loopback, Local Loopback, Automatic Echo

7.8.8 Serial Synchronous Controller

- Provides serial synchronous communication links used in audio and telecom applications (with CODECs in Master or Slave Modes, I2S, TDM Buses, Magnetic Card Reader, etc.)
- Contains an independent receiver and transmitter and a common clock divider
- Offers a configurable frame sync and data length
- Receiver and transmitter can be programmed to start automatically or on detection of different event on the frame sync signal
- Receiver and transmitter include a data signal, a clock signal and a frame synchronization signal

7.8.9 AC97 Controller

- Compatible with AC97 Component Specification V2.2
- Capable to Interface with a Single Analog Front end
- Three independent RX Channels and three independent TX Channels
  - One RX and one TX channel dedicated to the AC97 Analog Front end control
  - One RX and one TX channel for data transfers, connected to the DMACA
  - One RX and one TX channel for data transfers, connected to the DMACA
- Time Slot Assigner allowing to assign up to 12 time slots to a channel
- Channels support mono or stereo up to 20 bit sample length - Variable sampling rate AC97 Codec Interface (48KHz and below)

•

7.8.17 Image Sensor Interface

- ITU-R BT. 601/656 8-bit mode external interface support
- Support for ITU-R BT.656-4 SAV and EAV synchronization
- Vertical and horizontal resolutions up to 2048 x 2048
- Preview Path up to 640*480
- Support for packed data formatting for YCbCr 4:2:2 formats
- Preview scaler to generate smaller size image 50
- Programmable frame capture rate

Before a PWM period has elapsed, the read channel status may be wrong. The CHIDx-bit for a PWM channel in the PWM Enable Register will read '1' for one full PWM period even if the channel was disabled before the period elapsed. It will then read '0' as expected.

Fix/Workaround

Reading the PWM channel status of a disabled channel is only correct after a PWM period

14. **TWI transfer error without ACK**

If the TWI does not receive an ACK from a slave during the address+R/W phase, no bits in the status register will be set to indicate this. Hence, the transfer will never complete.

Fix/Workaround

To prevent errors due to missing ACK, the software should use a timeout mechanism to terminate the transfer if this happens.

15. **SSC can not transmit or receive data**

The SSC can not transmit or receive data when CKS = CKDIV and CKO = none in TCMR or RCMR respectively.

Fix/Workaround

Set CKO to a value that is not "None" and enable the PIO with output driver disabled on the TK/RK pin.

16. **USART - RXBREAK flag is not correctly handled**

The FRAME_ERROR is set instead of the RXBREAK when the break character is located just after the STOP BIT(S) in ASYNCHRONOUS mode.

Fix/Workaround

The transmitting UART must set timeguard greater than 0.

17. **USART - Manchester encoding/decoding is not working.**

Manchester encoding/decoding is not working.

Fix/Workaround

Do not use manchester encoding.

18. **SPI - Disabling SPI has no effect on TDRE flag.**

Disabling SPI has no effect on TDRE whereas the write data command is filtered when SPI is disabled. This means that as soon as the SPI is disabled it becomes impossible to reset the TDRE flag by writing in the SPI_TDR. So if the SPI is disabled during a PDC transfer, the PDC will continue to write data in the SPI_TDR (as TDRE keeps High) till its buffer is empty, and all data written after the disable command is lost.

Fix/Workaround

Disable PDC, 2 NOP (minimum), Disable SPI. When you want to continue the transfer: Enable SPI, Enable PDC.

19. **SPI disable does not work in SLAVE mode.**

SPI disable does not work in SLAVE mode.

Fix/Workaround

Read the last received data, then perform a Software Reset.

Configure the MCI Mode Register (MR) to accept 8-bit data input by writing a 1 to bit 13 (FBYTE), and transfer each byte of the transmit data to TDR by right aligning the useful value. This allows the number of bytes transferred into the TDR to match the number set up in the BCNT field of the MCI Block Register (BLKR).

35. Unreliable branch folding

In certain situations, branch folding does not work as expected.

Fix/Workaround

Write 0 to CPUCR.FE before executing any branch instructions after reset.

36. USB PLL jitter may cause packet loss during USB hi-speed transmission

The USB Hi-speed PLL accuracy is not sufficient for Isochronous USB hi-speed transmission and may cause packet loss. The observed bit-loss is typically < 125 ppm.

Fix/Workaround

Do not use isochronous mode if absolute data accuracy is critical.

10.2 Rev. B

Not sampled.

10.3 Rev. A

Not sampled.

11. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

11.1 Rev. F 09/09

1. Updated"Errata" .

11.2 Rev. E 09/09

1. Updated "Errata" .

11.3 Rev. D 09/07

1. PIO Controller C Multiplexing table updated in "Peripherals" on page 75".
2. Added section "USBA" on page 81 in Clock Connections in "Peripherals" on page 75.
3. USBA feature list updated in "Peripherals" on page 75.
4. Renamed clk_slow to clk_osc32 in Table 9-4 on page 80.
5 Updated organisation of User Interface in "HSB Bus Matrix (HMATRIX)" on page 142.
6. Updated special bus granting mechanism in "HSB Bus Matrix (HMATRIX)" on page 142.
7. Added product dependencies in "DMA Controller (DMACA)" on page 174.
8. Added product dependencies in "Peripheral DMA Controller (PDC)" on page 233.
9. Added description of multi-drive in "Parallel Input/Output Controller (PIO)" on page 249.
10 Added MDER/MDDR/MDSR to pin logic diagram in "Parallel Input/Output Controller (PIO)" on page 249.
11. SPI pins must be enabled to use local loopback.
12. Updated description of the OVRES bit in "SPI Status Register" on page 310.
13. Updated bit description of TXEMPTY in the "USART Channel Status Register" on page 432.
14. Number of chip select lines updatedin figures and tables, changed from 8 to 6 in "Static Memory Controller (SMC)" on page 489.
15. Made the MDR register Read/Write instead of Read in "SDRAM Controller (SDRAMC)" on page 532.
16. Removed the PWSEN and PWSDIS bits from the "MCI Control Register" on page 585.
17. Added PDCFBYTE and removed the PWSDIV bits from the "MCI Mode Register" on page 586.
18. Added note about reading the Status Register clears the interrupt flag in "Timer/Counter (TC)" on page 740.
19. Added debug operation to product dependencies in "Timer/Counter (TC)" on page 740.

20. Added debug operation to product dependencies in ”Pulse Width Modulation Controller (PWM)” on page 774.
21. Consistently used the term LCDC Core Clock through the document when referring to the generic clock that drives the LCD Core and is used to generate PCLK and the other LCD synchronization signals.
22. Updated typos in ”LCD Controller (LCDC)” on page 800.
23. Rewritten the Register Configuration Guide and renamed it “Register Configuration Example“ in ”LCD Controller (LCDC)” on page 800.
24. Updated formula for pixel clock in ”LCD Control Register 1” on page 840.
25. Updated HOZVAL description in ”LCD Frame Configuration Register” on page 845.
26. Updated ”PLL Characteristics” on page 933.
27. Updated ”Errata” on page 40.

11.4 Rev. C 07/07

1. Updated ”Part Description” on page 2.
2. PC Signals removed in ”Signals Description” on page 5
3. USB Signals updated in ”Signals Description” on page 5.
4. The PX0 - PX53 Signals added in ”Signals Description” on page 5.
5. SDCS signals removed from PIO Controller Multiplexing tables in ”Peripherals” on page 79.
6. MAC references removed form tables in ”Memories” on page 77.
7. MAC controller references removed in ”Peripheral overview” on page 94.
8. SDCS1 signal removed from figures and tables, and SDCS0 renamed to SDCS in ”External Bus Interface (EBI)” on page 147.
9. SmartMedia renamed to NAND Flash in some description to avoid confusion in ”External Bus Interface (EBI)” on page 147.
10. Updated Application block diagram in Figure 1-2 on page 1.
11. Updated the reset state of the SMC Mode register in Table 27-9 on page 523.
12. Updated ”Mechanical Characteristics” on page 927.
13. Updated pad parameters in ”DC Characteristics” on page 928.
14. Updated ”Power Consumption by Peripheral in Active Mode” on page 930, LCD and MACB excluded.
15. Updated pad parameters in ”Clock Characteristics” on page 931.
16. Updated ”USB Transceiver Characteristics” on page 934.
17. Updated ”EBI Timings” on page 939.

11.5 Rev. B 04/07

1. Updated ”Features” on page 1.
2. Updated tables in ”Signals Description” on page 4.
3. Updated Table 9-2 on page 77, Table 9-9 on page 82, and Table 9-10 on page 83 in the ”Peripherals” on page 75.
4. Updated module names and abbreviations through the datasheet.