Features

- Advanced RISC Architecture, 130 Powerful Instructions, Most Single-Clock Cycle Execution
- Clock Generator Provides CPU Rates up to 48 MHz
- Only One External Clock Crystal of 12 MHz Can Generate All the Required System Clocks:
  - Internal Clock for Standard UART Rates
  - A 48 MHz and 96 MHz Clock for USB Data Recovery
  - AVR Processor and System Clock
- Full-speed USB Interface (12 Mbits per Second) 2.0 Compliant
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-Scan Capabilities According to the JTAG Standard
  - Extensive On-Chip Debug Support
- Two On-chip 16550 UARTs Supporting Baud Rates up to 921 Kbaud
  - Both UARTs Incorporate Individual Transmit and Receive FIFOs of 16 Bytes
  - UART0 Supports Modem Control Signals
- Programmable SPI Interface
- On-chip Bootstrap ROM Provides a Variety of Firmware Upgrade Modes
  - Device Firmware Upgrade Through USB for the Internal Program SRAM (No External Non-volatile SPI Memory Required)
  - Device Firmware Upgrade Through USB for both the Internal Program SRAM and the External SPI DataFlash® or EEPROM
  - SPI Program Mode from the External DataFlash or EEPROM
- External Memory Interface Supporting up to 32 Kbytes of External RAM in Address Multiplexed Mode, 2 Banks of 256 Bytes in Non-multiplexed Mode, FIFO, or with an Extra 20 GPIOs
- DMA Channels Allow Fast Data Transfers between Endpoint Buffers and Internal or External SRAM (DMA Transfer Rate is 12 MHz for All Channels)
- 8K x 16 bits (up to 11K x 16 Bits), In-System SRAM for Program Code (Program Memory)
- On-chip 8 Kbytes SRAM for Data and Variables (2, 4, or 6 Kbytes can be Remapped for Program Storage in the Address Area Above the Program Memory
- Two 8-bit Timer/Counters
- One 16-bit Timer/Counter
- Four External Interrupts Through GPIOs
- Programmable Watchdog Timer
- Low Voltage Operation:
  - 1.8V for the Core
  - 1.8V or 3.3V for the Periphery
  - 3.3V for the USB
- 100-pin TQFP Package
- Applications
  - Programmable USB-to-Serial Bridge for RS-2332 Devices (Cell Phones, Printers, PDAs, etc.)
  - IrDA Control over USB
  - USB Memory Sticks
  - General High-speed Microcontroller Application

High-speed (48 MHz) AVR® Microcontroller with USB Interface

AT76C713

5665B–USB–04/05

1. Overview

The AT76C713 is a low-power, high performing USB 2.0 full-speed microcontroller providing advanced features for USB peripherals. It combines a number of functions required in such a device, including the following:

The device is based on the AVR-enhanced RISC architecture core, which combines an advanced instruction set with 32 general-purpose working registers. By executing powerful instructions in a single clock cycle, the AT76C713 achieves throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize the power consumption versus the processing speed

The clock generation circuit requires a clock input of 12 MHz and provides standard clock rates for the USB module and the on-chip UARTs, as well as several AVR CPU rates varying from 16 MHz up to 48 MHz

Internal DMA channels allow fast data transfers between the USB buffers and the external or the on-chip memory without processor interruption. USB DMA transfers use devoted data paths with a 12 Mbytes transfer rate

An on-chip flexible memory controller allows dynamic memory mapping and provides the required timing for interfacing with slow or fast external memory devices, like SRAM or FIFOs

Five multipurpose I/O ports, PORT(A-E), provide the signals for all the serial and parallel interfaces. Programmable strobe signals are provided for external FIFO access. In addition, the AT76C713 supports various power-down modes and offers four external interrupts, a programmable Watchdog Timer, and flexible Timer/Counters with compare modes

On power-up, the bootstrap code is executed from the boot ROM. The purpose of the bootstrap code is to load the application code into the program memory. The application code is executed from the on-chip SRAM program memory, contributing to the low-power consumption. Different programming modes are supported, depending on the application (that is, the mode is selected externally by the PMODE0 and PMODE1 pins)

In the slave programming mode, an external system (that is, the Host), operating as SPI master, can transfer the program image in a raw format to the program memory of the device. In this case, the AT76C713 operates as an SPI slave and starts running from the internal boot ROM code, which switches to the start of program memory when it detects the end of a valid program transfer from the Host to the AT76C713

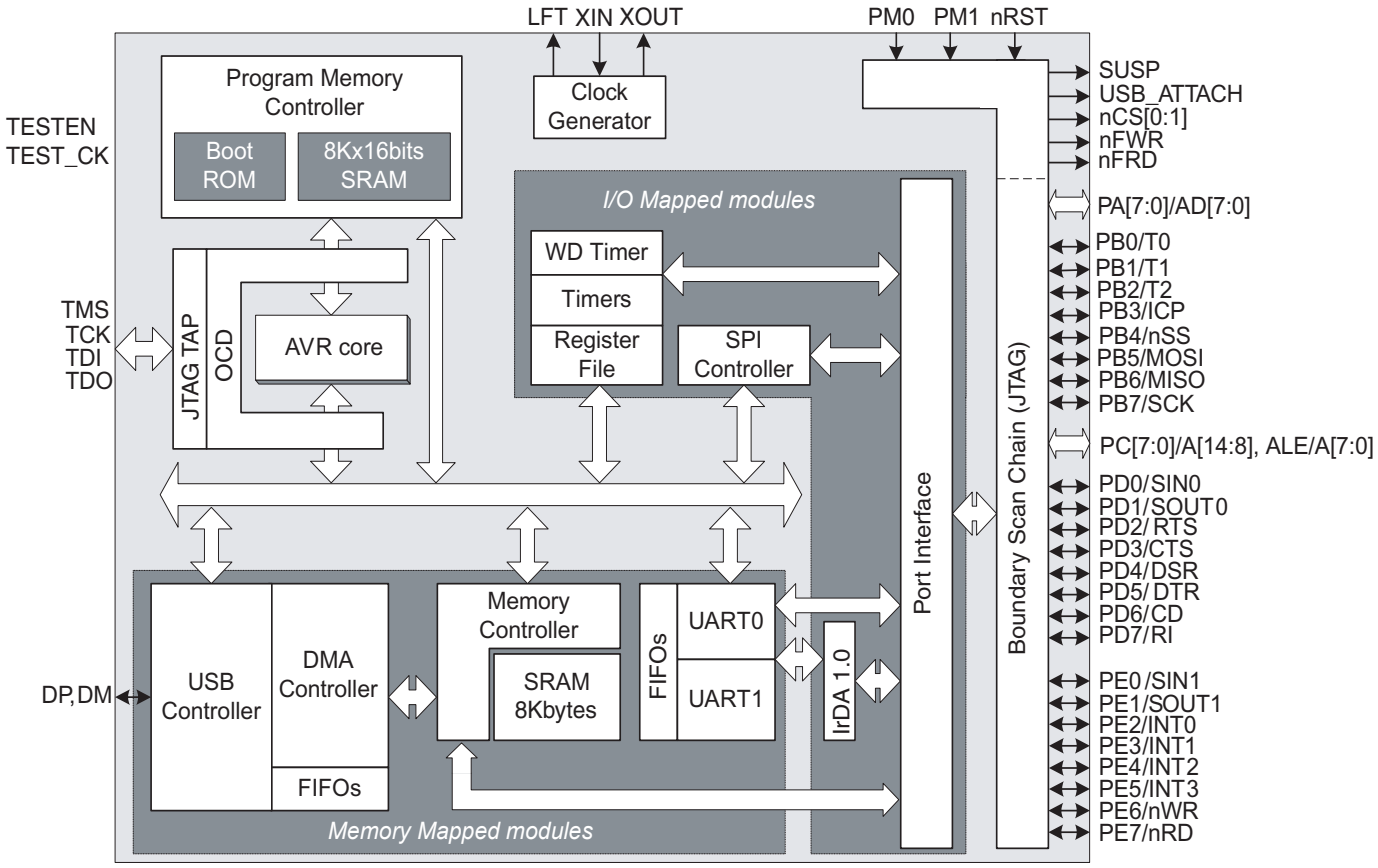
In the master programming mode, the AT76C713 reads the whole program image from an external serial EEPROM or DataFlash® and switches to the start of the program memory when it completes this reading. Alternatively, the AT76C713 reads only configuration parameters from a small serial non-volatile memory (EEPROM or DataFlash), enables the USB Controller, and executes the USB Device Firmware Upgrade (DFU) code that is stored in the boot ROM

The USB Controller consists of a Serial Interface Engine (SIE), a Function Interface Unit (FIU), and a System Interface (SI). The SIE performs bit processing, line coding, packet generation, packet type recognition, serial-parallel data conversion, and packet delineation. The FIU consists of a protocol engine and a USB device with one Control Endpoint (EP0) and four programmable Endpoints with up to 512 bytes maximum total size. All Endpoints support double buffering in order to provide the maximum performance specified for the USB

The AT76C713 supports two 16550 UART modules with 16 bytes FIFOs in each direction. The UART0 serial interface provides full modem control functionality with the RTS/CTS, DTR/DSR, RI, and CD signals. These signals are provided by the general-purpose I/O pins of PORTD

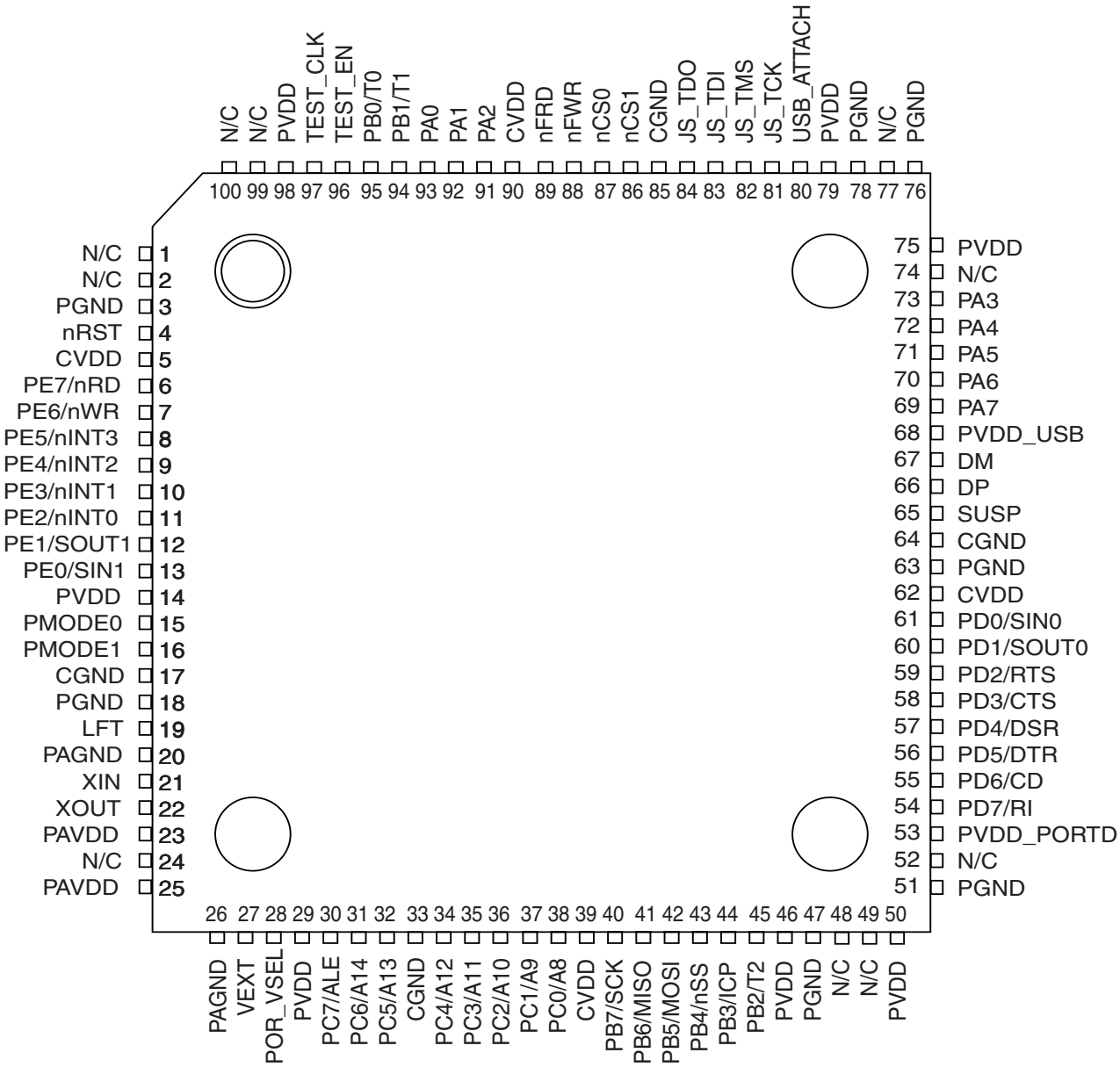
The AT76C713 AVR is supported with a full suite of program and system development tools, including: C compiler, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits

2. AT76C713 Functional Diagram



3. Pin Diagram

3.1 100-pin TQFP Package



4. Pin Summary – Pin Assignment

Table 4-1. Pin Summary – Pin Assignment

Pin #	Pin Name	Pin #	Pin Name	Pin #	Pin Name
1	N/C	35	PC3/A11	69	PA7
2	N/C	36	PC2/A10	70	PA6
3	PGND	37	PC1/A9	71	PA5
4	nRST	38	PC0/A8	72	PA4
5	CVDD	39	CVDD	73	PA3
6	PE7/nRD	40	PB7/SCK	74	N/C
7	PE6/nWR	41	PB6/MISO	75	PVDD
8	PE5/nINT3	42	PB5/MOSI	76	PGND
9	PE4/nINT2	43	PB4/nSS	77	N/C
10	PE3/nINT1	44	PB3/ICP	78	PGND
11	PE2/nINT0	45	PB2/T2	79	PVDD
12	PE1/SOUT1	46	PVDD	80	USB_ATTACH
13	PE0/SIN1	47	PGND	81	TCK
14	PVDD	48	N/C	82	TMS
15	PMODE0	49	N/C	83	TDI
16	PMODE1	50	PVDD	84	TDO
17	CGND	51	PGND	85	CGND
18	PGND	52	N/C	86	nCS1
19	LFT	53	PVDD_PORTD	87	nCS0
20	PAGND	54	PD7/RI	88	nFWR
21	XIN	55	PD6/CD	89	nFRD
22	XOUT	56	PD5/DTR	90	CVDD
23	PAVDD	57	PD4/DSR	91	PA2
24	N/C	58	PD3/CTS	92	PA1
25	PAVDD	59	PD2/RTS	93	PA0
26	PAGND	60	PD1/SOUT0	94	PB1/T1
27	VEXT	61	PD0/SIN0	95	PB0/T0
28	POR_VSEL	62	CVDD	96	TEST_EN
29	PVDD	63	PGND	97	TEST_CLK
30	PC7/ALE	64	CGND	98	PVDD
31	PC6/A14	65	SUSP	99	N/C
32	PC5/A13	66	DP	100	N/C
33	CGND	67	DM		
34	PC4/A12	68	PVDD_USB		

5. Signal Description

Table 5-1. Signal Description

Type: I = Input, O = Output, I/O = Bi-directional, A = Analog Signal

Name	Type	Description
JTAG Signals		
TCK	I	JTAG clock input
TDI	I	Scan chain bitstream input
TDO	O	Scan chain bitstream output
TMS	I	JTAG mode select input
Port Signals		
PA[0:7]	B	Port A is used in configurations with external memory, where it acts as the AD0-7 address and/or data bus. If the alternative function of the port is not used, Port A also serves as an 8-bit bi-directional I/O port with internal pull-up resistors. Port A output buffers can sink 20 mA and can drive the LED displays directly. Port A pins that are externally pulled low will source current. The Port A pins are inputs when a reset condition becomes active, even if the clock is not running.
PB[0:7]	B	The high nibble of port B (PB[7–4]) offers the signals for the SPI interface. The low-nibble (PB[3-0]) is used from the three internal timers as trigger and capture inputs. These functions can operate regardless the direction of the PB[3-0] pins. Port B serves also as an 8-bit bi-directional I/O port with internal pull-up resistors, if the alternative functions of the port are not used. The Port B output buffers can sink 20 mA and can drive LED displays directly. As inputs, Port B pins that are externally pulled low will source current. The Port B pins are inputs when a reset condition becomes active, even if the clock is not running. The alternative functions of Port B are explained in more details at the following table and at the proper sections.
PC[0:7]	B	Port C is used in configurations with external memory, where it acts as the address bus bits A8-15 and ALE in a multiplexed address mode and A0-7 in a non-multiplexed mode. If the alternative functions of the port are not used, Port C also serves as an 8-bit bi-directional I/O port with internal pull-up resistors.The Port C output buffers can sink 20 mA and can drive the LED displays directly. Port C pins that are externally pulled low will source current. The Port C pins are inputs when a reset condition becomes active, even if the clock is not running.

Alternative Functions of Port B

Port	Alternate Function
PB0	T0: Timer/Counter 0 clock input
PB1	T1: Timer/Counter 1 clock input
PB2	T2: Timer/Counter 2 clock input
PB3	ICP1: Input Capture Pin for Timer/Counter1
PB4	nSS: SPI slave port select input
PB5	MOSI: SPI slave port select input
PB6	MISO: SPI master data in, slave data out
PB7	SCK: SPI master clock out, slave clock in

Table 5-1. Signal Description (Continued)

Type: I = Input, O = Output, I/O = Bi-directional, A = Analog Signal

<table>
<tr><th>Name</th><th>Type</th><th>Description</th></tr>
<tr><td>PD[0:7]</td><td>B</td><td>Port D offers the data and handshaking signals for UART0 interface, as listed below.
When the UART0 is not used, PORT D is also an 8-bit bi-directional I/O port with internal pull-up resistors. The Port D output buffers can sink 20 mA. Port D pins that are externally pulled low will source current. The Port D pins are inputs when a reset condition becomes active, even if the clock is not running.

Alternative Functions of Port D
<table><tr><th>Port</th><th>Alternate Function</th></tr><tr><td>PD0</td><td>SIN0, Serial In UART0 (I): This pin provides the serial receive data input to UART0</td></tr><tr><td>PD1</td><td>SOUT0, Serial Out UART0 (O): This pin provides the serial transmit data from the UART0</td></tr><tr><td>PD2</td><td>nRTS, Ready To Send (B): RTS UART0 handshaking signal</td></tr><tr><td>PD3</td><td>nCTS, Clear To Send (B): CTS UART0 handshaking signal</td></tr><tr><td>PD4</td><td>nDSR, Data Set Ready (B): DSR UART0 handshaking signal</td></tr><tr><td>PD5</td><td>nDTR, Data Terminal Ready (B): DTR UART0 handshaking signal</td></tr><tr><td>PD6</td><td>nCD, Carrier Detect (B): CD UART0 handshaking signal</td></tr><tr><td>PD7</td><td>nRI, Ring Indicator (B): RI UART0 handshaking signal</td></tr></table></td></tr>
<tr><td>PE[0:7]</td><td>B</td><td>Port E offers the data signals for the UART1 interface, offers 4 external interrupt lines (edge triggered or level sensitive external interrupt) and the nWR/nRD signals for configurations with external memory.
If the alternative functions are not used, PORT E serves as an 8-bit bi-directional I/O port with internal pull-up/down resistors. Pins PE2 and PE3 have pull-down resistors while the rest pins have pull-up resistors. The output buffers of Port E pads can sink 20 mA. Port E pins that are externally pulled low (or high for PE2 and PE3 pins) will source current. The Port E pins are inputs when a reset condition becomes active, even if the clock is not running.

Alternative Functions of Port E
<table><tr><th>Port</th><th>Alternate Function</th></tr><tr><td>PE0</td><td>SIN1, Serial In UART1 (I): This pin provides the serial receive data input to UART1.</td></tr><tr><td>PE1</td><td>SOUT1, Serial Out UART1 (O): This pin provides the serial transmit data from the UART1</td></tr><tr><td>PE2</td><td>INT0, edge-triggered or level sensitive interrupt with pull-down</td></tr><tr><td>PE3</td><td>INT1, edge-triggered or level sensitive interrupt with pull-down</td></tr><tr><td>PE4</td><td>INT2, edge-triggered or level sensitive interrupt with pull-up</td></tr><tr><td>PE5</td><td>INT3, edge-triggered or level sensitive interrupt with pull-up</td></tr><tr><td>PE6</td><td>nWR</td></tr><tr><td>PE7</td><td>nRD</td></tr></table></td></tr>
</table>

Table 5-1. Signal Description (Continued)

Type: I = Input, O = Output, I/O = Bi-directional, A = Analog Signal

Name	Type	Description
USB Serial Interface		
DP	I/O	USB data I/O (positive differential line). DP and DM form the differential signal pin pair connected to the Host Controller or an upstream Hub.
DM	I/O	USB data I/O (negative differential line)
USB_ATTACH	O	This output controls the pullup resistor of the DP signal. When low, it enforces the host to re-enumerate the device.
Programming Mode Control Signals		
PMODE0 PMODE1	I	PMODE0 and PMODE1 pins are used from the on-chip bootstrap code and define the programming mode. For more details refer to the Program Modes section.
Test Signals		
TEST_EN	I	General-purpose signal for test. Tied to V_{SS} in normal conditions.
TEST_CLK	I	Used only for production setting. Tied to V_{SS} in normal conditions.
POR_VSEL	I	Used in production phase only. Tied to V_{SS} in normal conditions.
VEXT	I	Used in production phase only. Tied to 1.8V in normal conditions.
Other Signals		
nRST	I	Reset input. A low on this pin for two clock cycles while the oscillator is running resets the device. Note that there is no internal pull-up resistor on this pin.
SUSP	O	Indicates if the IC is in power down mode
nCS0	O	External Memory Chip Select 0
nCS1	O	External Memory Chip Select 1
nFWR	O	External FIFO Write
nFRD	O	External FIFO Readl
XIN	I	System clock oscillator pad, input to the inverting oscillator amplifier and input to the internal system clock operating circuit.
XOUT	O	System clock oscillator pad, output from the inverting oscillator amplifier.
LFT	A	An external RC filter should be connected to this pin to stabilize the lock-in time of the internal PLL for the master clock input XIN.Main PLL LFT input
Power Supply pins		
CGND	Power	0 Volts supply to the core
CVDD	Power	1.8 Volts supply to the core
PGND	Power	0 Volts supply to the external section of the I/O circuitry
PVDD	Power	1.8V or 3.3 Volts supply to the external section of the I/O circuitry
PVDD_PortD	Power	1.8V or 3.3 Volts supply to the external section of the Port D I/O circuitry
PVDD_USB	Power	3.3 Volts supply for the USB and SUSPEND pins
PAGND	Power	0 Volts Analog supply to the core, AC and DC sections of the I/O circuitry
PAVDD	Power	1.8 Volts Analog supply to the core, AC and DC sections of the I/O circuitry

6. Functional Description

6.1 Bootstrap ROM and Programming Modes

The bootstrap code in the boot ROM of the AT76C713 is always activated on power-up or after a system reset. The functionality of the bootstrap code is configured by pins PMODE0 and PMODE1. In the SPI master programming modes, the AT76C713 is master over the SPI interface and can support an external EEPROM or DataFlash. The SPI slave programming mode is useful in a system where an external host is present and keeps the image code for the AT76C713 in its own memory.

Table 6-1. Programming Modes

PMODE0	PMODE1	Programming Mode
0	0	USB DFU only (SPI disabled)
0	1	SPI Slave: SPI master host must be present
1	0	SPI Master: SPI serial EEPROM is assumed
1	1	SPI Master: SPI DataFlash is assumed

6.1.1 USB DFU-only Mode

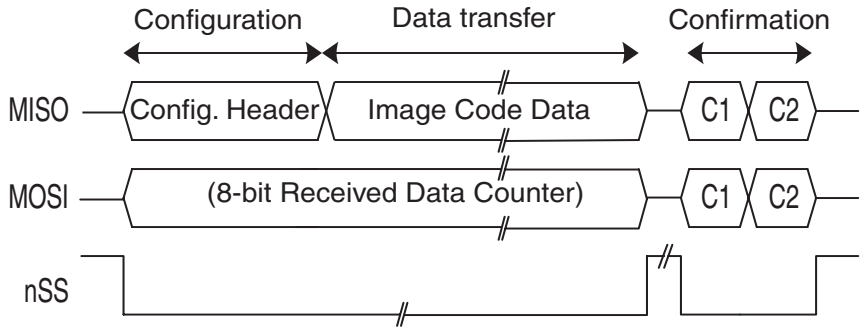
When both the PMODE0 and PMODE1 pins are tied low, the bootstrap code will not enable the USB controller but not the SPI controller. The system will wait for a USB DFU sequence using the default USB descriptors stored in the boot ROM.

6.1.2 Slave Programmming Mode

When the PMODE0 pin is tied low and the PMODE1 pin is tied high, the bootstrap ROM configures the SPI as a slave and waits for a specific sequence of data through the SPI. The sequence is as follows

1. The nSS pin is enabled.
2. The Configuration Header (256 bytes) is transmitted.
3. Image of the code to be stored in program memory is transmitted, starting from address 0. (The code size is determined in the Configuration Header.)
4. The nSS pin is disabled and re-enabled.
5. Confirmation byte 1 is transmitted.
6. Confirmation byte 2 is transmitted.
7. The nSS pin is disabled

Figure 6-1. Slave Programming Mode Sequence



During the transmission of the configuration data and code image, the AT76C713 returns an 8-bit counter value that counts the received bytes. Thus, the Host can verify if the AT76C713 has received the correct amount of data.

During the first confirmation cycle, the SPI master sends $CA to command “Continue Analyzing” or $AB to command “Abort”. The SPI slave will return “Data was OK” by sending $D0 or “Error” by sending $EE.

During the second confirmation cycle, the SPI master data is ignored and the SPI slave sends the positive acknowledgment $AD “Analyzing Data”, or one of the following error messages: $DD “Discarding data”, $DE “Discarding due to transmission error”, or $EE “Error in confirmation command”.

Table 6-2. Possible Cases of Programming during the Slave Program Mode

Case		1	2	3	4
Cycle 1	Master	$CA	$AB	xx	Invalid
	Slave	$D0	$D0	$EE	xx
Cycle 2	Master	xx	xx	xx	xx
	Slave	$AD	$DD	$DE	$EE
Programming		OK	Fail	Fail	Fail

If the slave programming sequence fails, then the AT76C713 will enable the USB controller waiting for the DFU sequence, using the default USB descriptors.

6.1.3 Master Programming Modes

When the PMODE0 pin is tied high, the bootstrap ROM configures the SPI as a master and is prepared to communicate either with a serial EEPROM if PMODE1 is tied low, or with a DataFlash if PMODE1 is tied high. The AT76C713 supports any Atmel EEPROM of the AT25xxx type (for example, AT25040) and Dataflash of the AT45xxx type.

The first 256 bytes that are read by the EEPROM or DataFlash are known as the configuration header, which contains useful information for the rest of the downloading procedure.

The bytes fetched from the external serial memory may be an image code or configuration parameters. If the bytes are an image code, after data fetching is complete the program running in the boot ROM resets the AVR program counter and switches to the SRAM program memory by setting the remap bit in the memory access interface.

If the data fetched from the external serial memory is configuration parameters only, the bootstrap code of the boot ROM will not switch to the program memory at the end of data fetching. Instead, it will enable the USB controller. The USB will enumerate using the descriptors loaded from the SPI memory. Upon enumeration completion, if a DFU application is enabled in the host, then the bootstrap ROM will support program storing in the program memory through the USB. When program storing is complete, the device will switch to the program memory.

6.1.4 Configuration Header

The Configuration Header is at least 256 bytes length and consumes the first page of the SPI memory. For systems that use SPI slave programming or SPI EEPROM memory (for example, AT25128A), the Configuration Header length is fixed to 256 bytes. For systems that use SPI DataFlash, the page size is equal to the page size of the used DataFlash. For example, using the AT45DB011B, the page size is 264 bytes and thus, the Configuration Header will consume 264 bytes.

The meaning of each byte is described in Table 6-3.

Table 6-3. Configuration Header

Byte	Description	
0	$55	
1	$AA	
2	Size[7:0]	
3	Size[15:8]	
4	Control Byte	
5	MEMMAP	
6	CIS[ ]	Device descriptor
		Configuration descriptor
		Serial descriptor
		Product descriptor

The first 2 bytes must be $55 and $AA in order to detect the rest of the information as valid. If those bytes are invalid, then the bootstrap code rejects the operation immediately and starts the USB DFU procedures.

Size[15:0] is the length of the image code counted in pages. During the slave programming mode and the SPI EEPROM master programming mode, each page is 256 bytes long.

The control bitmap (or control byte) is described in Table 6-4.

Table 6-4. Control Bitmap

Control Byte		
Bit	Field	Description
7	WSOF	Wait USB SOF.
6	—	Reserved to zero (‘0’).
5	PLCK	If set, then don’t use the PLL Lock signal (see also the PLCK bit of “Clock Control Register (CLK_CNTR)” on page 51).

Table 6-4. Control Bitmap

Control Byte		
Bit	Field	Description
4	REMAP	If set, then remap right after the downloading of the image code is successful.
3	STDBY	If set, use for sleep state in suspend mode the stand-by mode instead of power-down mode.
2	INTPROT	Interrupt Protect.
1	DTCH	Detach Enable: If set, detach before remap.
0	SUSP	If set the Suspend Circuit is on.

Note: Bits 7, 3, 2, 1 and 0 affect only the bootstrap USB DFU process

MEMMAP byte will be assigned immediately to the MEMMAP register (see MEMMAP description in the "Memory Access Interface" on page 15). This register configures the size of the program memory and data memory.

CIS[ ] is a table that carries the USB descriptors. Each descriptor starts right after the end of the previous one. The order is shown in Table 6-3. The remaining bytes from the end of the CIS[ ] table up to the end of the page have no meaning.

Note: If the Remap bit of the control byte is cleared, then the image code will be loaded into the AT76C713 program memory (see Size[15:0]), the system will not remap, and the bootstrap code will enable the USB core and enumerate using the new USB descriptors (CIS[ ]) instead of the default USB descriptors.

6.2 AVR Core

The AT76C713 chip is based on the AVR core architecture. All peripherals, apart from SPI, IrDA, and timers, are memory mapped to the data address space.

6.2.1 Interrupt Handling

The interrupt vector table of the AT76C713 is shown in Table 6-5.

Table 6-5. Interrupt Vector Table of the AT76C713

Vector Number	Program Address	Source	Interrupt Definition
1	$0000	RESET	H/W Pin and Watchdog Reset
2	$0002	SUSP_RESM	USB Suspend and Resume
3	$0004	USB	USB H/W & USB DMA Interrupt
4	$0006	INT0	External Interrupt Request 0
5	$0008	INT1	External Interrupt Request 1
6	$000A	INT2	External Interrupt Request 2
7	$000C	INT3	External Interrupt Request 3
8	$000E	TIMER2_OVF	Timer/Counter2 Overflow
9	$0010	TIMER1_CAPT	Timer/Counter1 Capture Event
10	$0012	TIMER1_COMPA	Timer/Counter1 Compare Match A
11	$0014	TIMER1_COMPB	Timer/Counter1 Compare Match B

Table 6-5. Interrupt Vector Table of the AT76C713

Vector Number	Program Address	Source	Interrupt Definition
12	$0016	TIMER1_OVF	Timer/Counter1 Overflow
13	$0018	TIMER0_OVF	Timer/Counter0 Overflow
14	$001A	SPI_STC	SPI Serial Transfer Complete
15	$001C	UART0_IRQ	UART0 Interrupt Request
16	$001E	UART1_IRQ	UART1 Interrupt Request

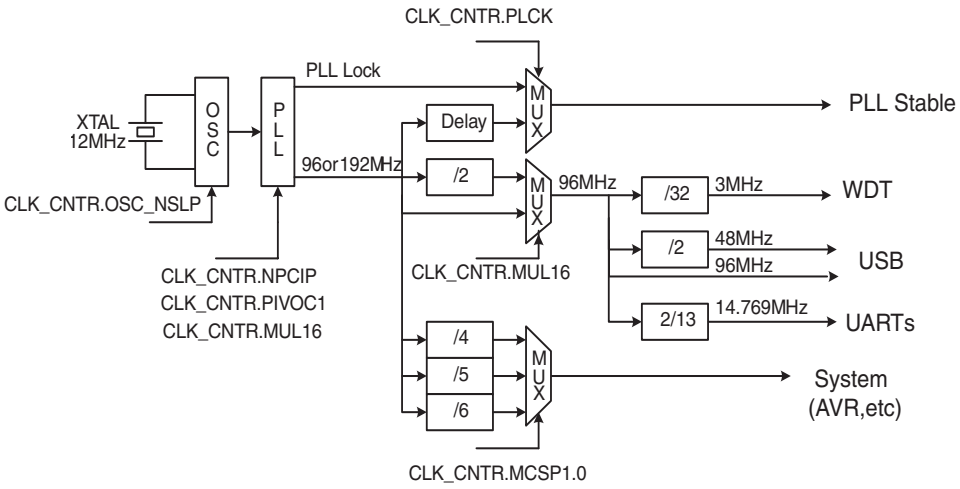
Note: The program memory word length is 16-bits wide.

6.3 Oscillator and Clock Generator

The AT76C713 clock generation circuit is based on a single clock input from an external 12 MHz crystal connected to the oscillator pad. All internal clocks are generated by multiplying this clock input incorporating an on-chip PLL. The output of the PLL can be configured in 96 MHz (default) or 192 MHz. In order to produce the desired clocks, the clock generator circuit divides the PLL output. Generating and then dividing a high frequency PLL output reduces the impact of jitter imported by the PLL.

An elaborate gobbling circuit produces a 14.769 MHz clock from the PLL output, which can support all standard baud rates with an acceptable frequency error.

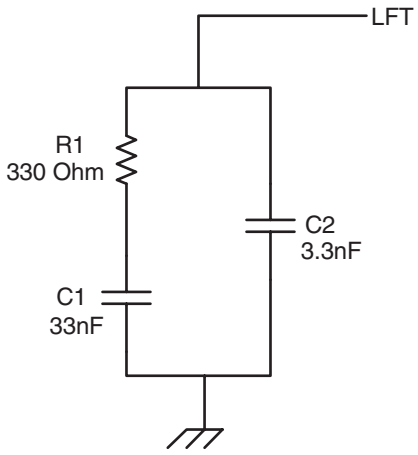
Figure 6-2. AT76C713 Clock Generation Tree



6.3.1 PLL LFT Pin

Figure 6-3 shows the circuit that must be attached to an LFT pin. The circuit consists of R1 = 330 Ω, C1 = 33nF, and C2 = 3.3nF.

Figure 6-3. LFT Pin Circuit



6.4 Memory Map

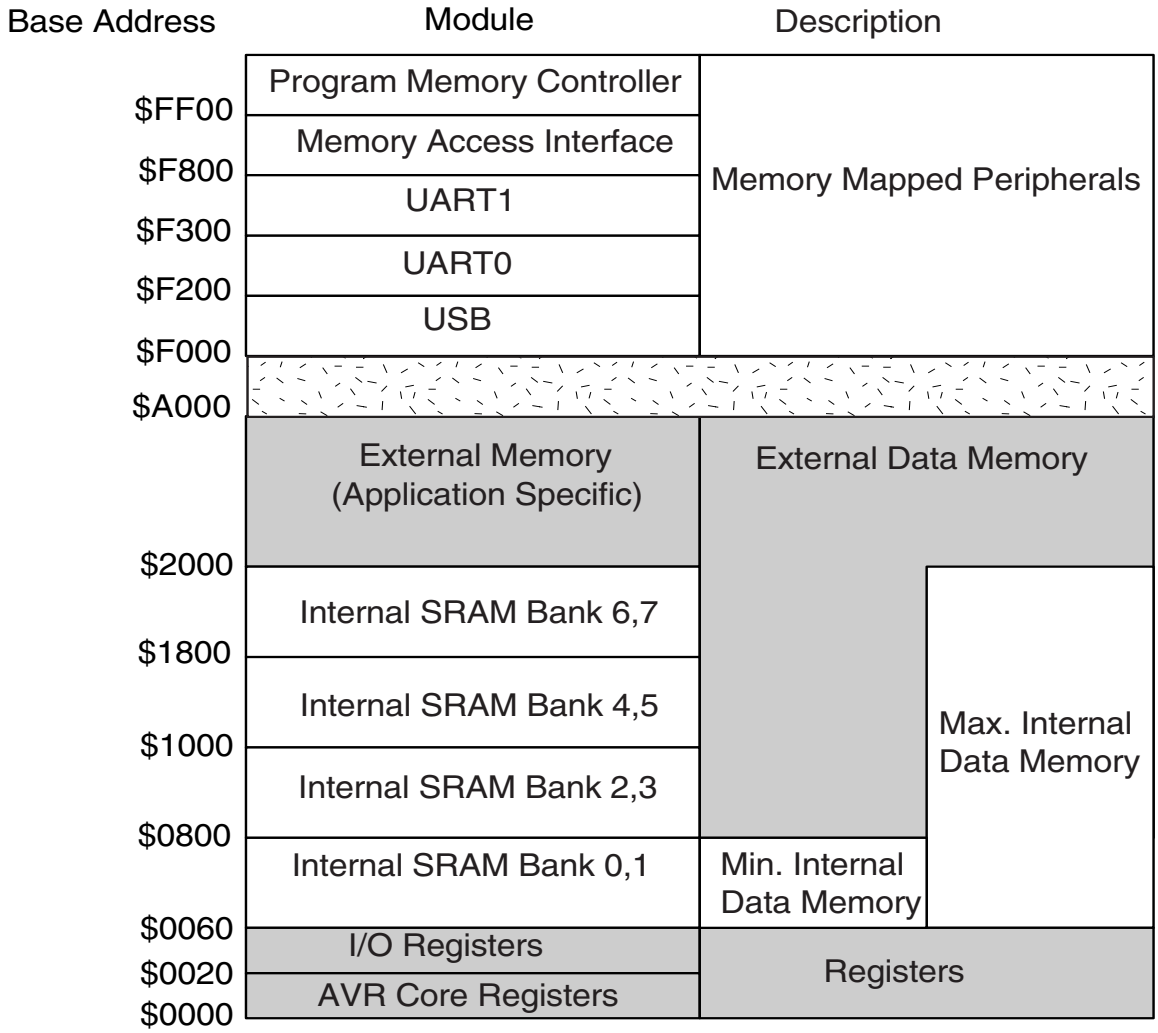


Table 6-6. Memory Map

Address Space	Size (Bytes)	Module
$FF00-$FF04	**5**	Program memory controller (PMC)
$F800-$F802	**3**	Memory access interface
$F300-$F30C	**13**	UART1
$F200-$F20C	**13**	UART0
$F000-$F0FD	**254**	USB
(End of Internal SRAM)-$7FFF	**(Application specific)**	External SRAM
$1800-$1FFF	**2048**	Internal SRAM bank 3
$1000-$17FF	**2048**	Internal SRAM bank 2
$0800-$0FFF	**2048**	Internal SRAM bank 1
$0060-$07FF	**2048**	Internal SRAM bank 0
$0020-$005F	**64**	I/O registers
$0000-$001F	**32**	AVR registers

6.5 Memory Access Interface

The memory access interface consists of the memory remapping interface and the external memory interface.

6.5.1 Memory Remapping Interface

On power-up, the device has 11K words (22K x 8) of program SRAM and 2 Kbytes of Data SRAM.

Utilizing the MEMMAP register (allocated in $F800 Data Memory address), it is possible to resize the two memory spaces. The last three 2-Kbyte banks of program memory can be reallocated and used as data memory and vice versa. According to Table 6-7, there are three possible options:

1. To remap the last 2 Kbytes (data banks 6 and 7)
2. To remap the last 4 Kbytes (data banks 4 to 7)
3. To remap the last 6 Kbytes (data banks 2 to 7)

Figure 6-4 and Table 6-7 illustrate the remapping.

Figure 6-4. Memory Remapping

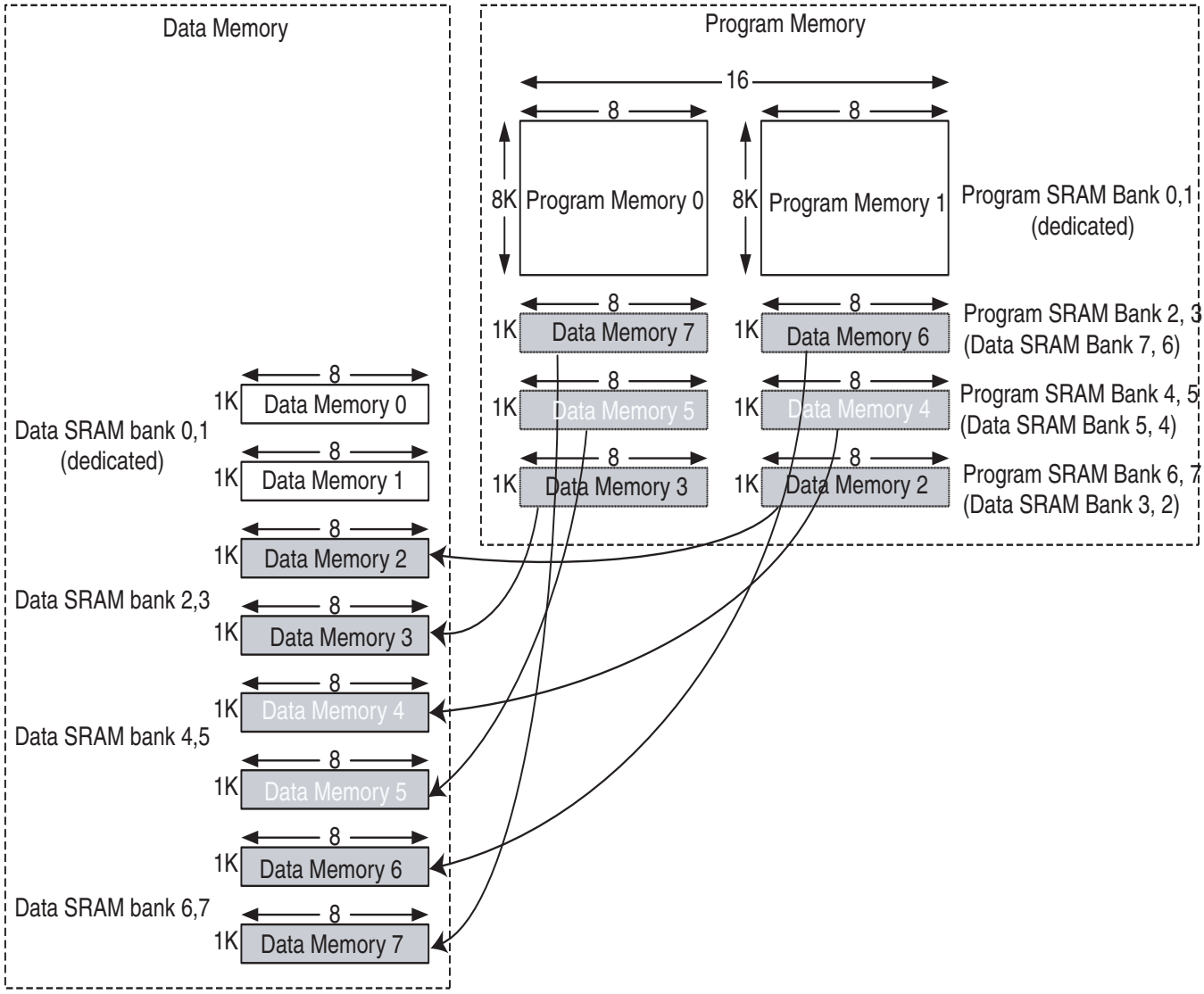


Table 6-7. Memory Map Allocation Cases

Data Banks Allocated to Data Memory Space	MEMMAP Register value	Data Address Space	Program Address Space
0	1	$0000-$07FF (2 Kbytes)	$8000-$D7FF (22 Kbytes)
2	3	$0000-$0FFF (4 Kbytes)	$8000-$CFFF (20 Kbytes)
4	5	$0000-$17FF (6 Kbytes)	$8000-$C7FF (18 Kbytes)
6	7	$0000-$1FFF (8 Kbytes)	$8000-$BFFF (16 Kbytes)

6.5.2 External Memory Interface (EMI)

The external memory can have various configurations and can be accessed by either the AVR core or the USB DMA Controller (see also the “USB DMA Controller” on page 19).

The PERIPHEN I/O register (EMIEN bit) (see description on page 52) enables and disables the external memory interface.

The registers that control the external memory access by the AVR core are located in the I/O space and named EMICRA and EMICRB.

The registers that control the external memory access by the USB DMA controller are located in the memory space and named DMA_EMICRA, DMA_EMICRB.

Table 6-8. The EMI Registers

Register	Offset Address	Function
EMICRA	$1C ($3C)	Controls the external memory access by the AVR core
DMA_EMICRA	($F801)	Controls the external memory access by the USB DMA controller
EMICRB	$1D ($3D)	Controls the external memory access by the AVR core
DMA_EMICRB	($F802)	Controls the external memory access by the USB DMA controller

The bit position of the two register pairs, EMICRA with DMA_EMICRA and EMICRB with DMA_EMICRB, are identical.

The EMD0/1 bits select the external memory interface mode, according to Table 6-9.

Table 6-9. EMDO/1 Bits

EMD1	EMD0	External Memory Interface Mode
0	0	FIFO
0	1	Reserved
1	0	Demultiplexed
1	1	Multiplexed

There are three control signals related to the control bits:

- The read signal (nRD or nFRD) controlled by RW0/1 and RM0/1 bits
- The write signal (nWR or nFWR) controlled by WW0/1 and WM0/1 bits
- The address latch enable signal (ALE) controlled by AW0/1 and AM0/1 bits

The RW0/1, WW0/1, and AW0/1 bits control the wait states inserted in the corresponding (read, write, and ALE) signals.

The RM0/1, WM0/1, and AM0/1 bits control the mode (waveform) of the corresponding (read, write, and ALE) signals.

Figure 6-5 shows the signal waveform as defined by the corresponding control bits xW0/1 and xM0/1, where ‘x’ can be ‘R’, ‘W’, or ‘A’ (for read, write, or ALE signal).

Figure 6-5. The Signal Waveform of Control Bits xW0/1, xM0/1

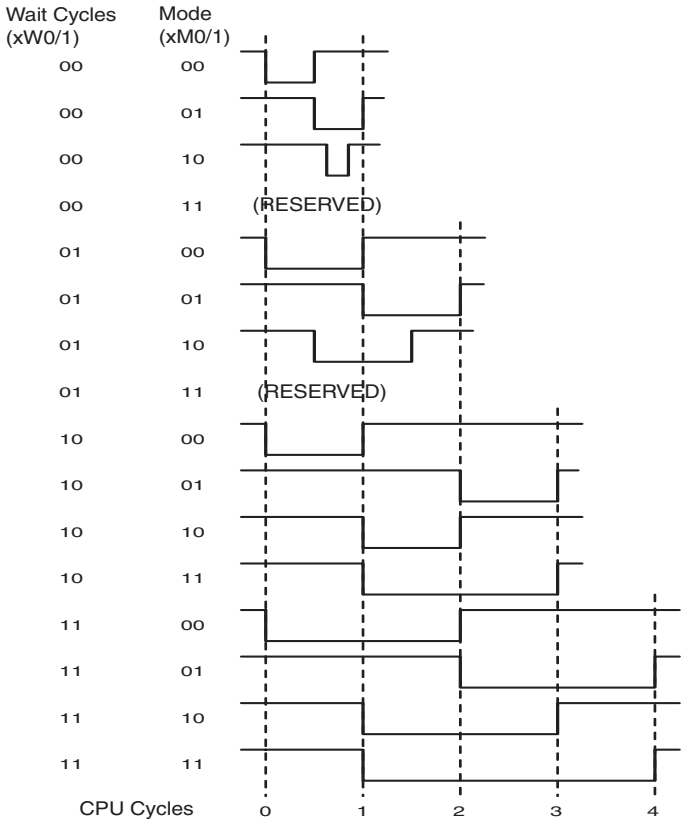


Table 6-10. The Functionality of the nCS0 and nCS1 Pins During an External Memory Access Cycle

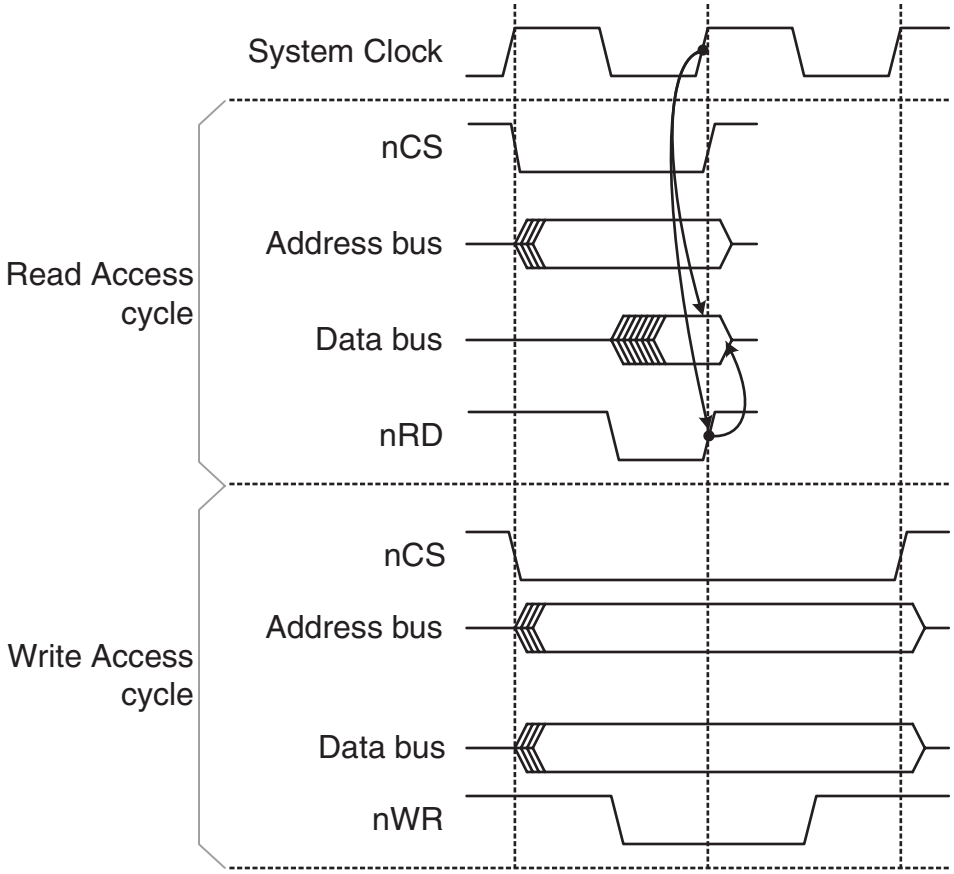
External Memory Interface Mode	ramadr[8]	nCS0	nCS1
FIFO	x	1	1
DeMUX	0	0	1
DeMUX	1	1	0
MUX	x	1	0

Note: “ramadr[8]” is the address bit 8 of the internal AVR data memory space.

The external memory address bus is equal to the internal AVR data address bus.The nCS0 (or nCS1) and the data bus are valid during the write cycle. During the read cycle, the nCS0 (or nCS1) is valid during the access cycle, while the data bus must be valid during the end of the cycle (rising edge of system clock, which causes the rising edge of nCS pin).

Figure 6-6 shows an example of the read and write accesses on a demultiplexed external memory (EMD1/0 = 10). The read cycle is defined with zero wait states (RW1/0 = 00) and waveform mode 01 (RM1/0 = 01). The write cycle is defined with one wait state (WW1/0 = 01) and waveform mode 10 (WM1/0 = 10).

Figure 6-6. Example of Read and Write Access on a Demultiplexed External Memory



6.6 USB DMA Controller

The AT76C713 offers a flexible DMA mechanism for fast data transfers between the Endpoint buffers and the memory. Using DMA, the AVR program is not occupied with data transfers and this results in the conservation of processing time. The DMA controller is capable of moving data as fast as 12 Mbytes per second.

The USB DMA controller has direct access to either internal or external data memory. When the USB starts a DMA transfer to or from the data memory, then the DMA controller prevents the AVR from accessing the same data memory area (internal or external). During a DMA transfer, the AVR can access any of the general-purpose registers. Once the AVR performs an access to the data memory area where a DMA transfer is occupied, the DMA controller then imposes wait cycles to the AVR until the DMA transfer is complete. Thus, no further improvement is required from the firmware to prevent external or internal memory access during the DMA transfers.

To enable a DMA transfer, the following steps should be performed

1. Program the offset address of the FIFO Data register of the associated Endpoint (this is the address of the FDRx register, that is, $F0CE for FDR1)
2. Program the number of the bytes to be moved to/from the FIFO Data register
3. Enable the write or read DMA cycle

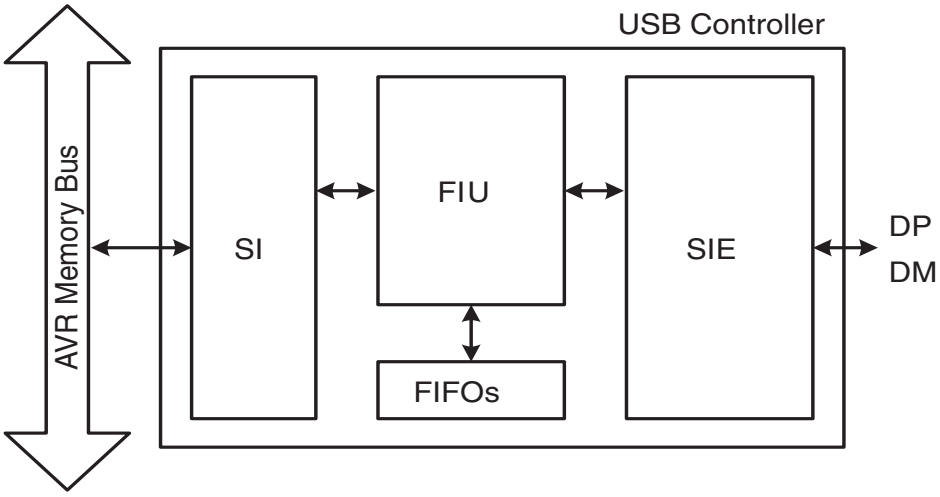
The AVR must poll the USB in order to detect the DMA completion.

6.7 USB Controller

The USB controller contains its own internal register file, which is memory mapped onto the AVR data bus. The USB block implements the low-level processing functions of the USB protocol stack described in Universal Serial Bus Specification, Version 1.1 and handles the data transfers between the Endpoint (EP) FIFOs and the system memory.

The USB controller consists of an SIE, an FIU, and the SI.

Figure 6-7. USB Controller



The USB controller supports one control Endpoint and seven programmable Endpoints in terms of the type (BULK, INT, or ISO) and the direction (IN, OUT). The FIU includes the Endpoint buffers and the controllers of Endpoints shown in Table 6-11.

Table 6-11. USB Function Interface Unit Structure

Endpoint	Type	Number of Bytes	Buffer Type
Endpoint 0	Control	16	Single
Endpoint 1..3	Any	128 (2 x 64)	Double
Endpoint 4	Any	64	Single
Endpoints 5..7	Any	0	Single

A Pair Endpoint address scheme is also supported. According to this scheme, two Endpoints may have the same address, provided one of the endpoints has been configured as IN and the other as OUT.

6.7.1 USB System Interface (SI)

The USB SI is necessary to bridge the AVR data bus with the USB controller because the system clock and the USB controller clock are asynchronous.

When the AVR accesses the USB registers, the USB system interface inserts wait cycles to the AVR core in order to synchronize the data memory bus with the USB local bus.

6.7.2 USB Interrupt Handling

All interrupt signals from the USB functions (protocol handler) are consolidated into the USB interrupt line. The interrupt line from the USB protocol handler is then multiplexed with the individual interrupt signals generated internally, and a single interrupt output is provided to the system interrupt controller.

All interrupts are masked through the interrupt enable registers that exist in the USB controller. The external resume and received resume interrupts are cleared when the firmware clears the interrupt bit. The suspend interrupt is automatically cleared when activity is detected. All other interrupts are cleared when the processor sets a corresponding bit in an Interrupt Acknowledge register in the USB macro cell. There is only one bit for each interrupt source.

Table 6-12 describes each of the interrupt sources of the USB protocol handler (see also the UISR register).

Table 6-12. USB Interrupt Sources

Interrupt	Description
Function EP0 interrupt	See control transfers at function EP0 for details
Function EP1 interrupt	For an OUT Endpoin,t it indicates that Function Endpoint1 has received a valid OUT packet and that the data is in the FIFO. For an IN Endpoint, it means that the Endpoint has received an IN token, sent out the data stored in the FIFO, and received an ACK from the Host. The FIFO is now ready to be written by new data from the processor
Function EP2 interrupt	See Function EP1 interrupt
Function EP3 interrupt	See Function EP1 interrupt
Function EP4 interrupt	See Function EP1 interrupt
Function EP5 interrupt	See Function EP1 interrupt
Function EP6 interrupt	See Function EP1 interrupt
SOF Received	The USB H/W decodes a valid Start of Frame
EXT RSM	The H/W has received a remote wake-up request
RCVD RSM	The H/W has received resume signaling. The processor's firmware should take the function out of the suspended state
SUSP	The H/W has detected a suspend condition and is preparing to enter the suspend mode. The processor's firmware should place the embedded function in the suspend mode

6.7.3 Interrupt Priority

The USB macro interrupt priority is defined in Table 6-13.

Table 6-13. USB Macro Interrupt Priority

Priority Level	Interrupt Name
2: High level	SOF Received
1: Same level (low level)	Function EP0 to EP6

6.7.4 Endpoint Interrupt

Endpoint interrupts are triggered by the setting or clearing of one or more bits in the Control and Status registers of an Endpoint. These interrupts are caused by events during packet transac-

tions and are different for Control and non-Control Endpoints. The interrupts are described below with respect to the Control and Status register bit definitions.

6.7.5 Interrupt for Non-control Endpoints

- RX OUT Packet set (0 -> 1)
- TX Packet Ready clear (1 -> 0)

6.7.6 Interrupt for Control Endpoints

- RX OUT Packet set (0 -> 1)
- RX SETUP set (0 -> 1)
- TX Packet Ready clear (1 -> 0)
- TX Complete set (0 -> 1)

6.7.7 USB Function Interface (FIU)

The FIU provides the interface between the processor (via the SI) and the SIE. It manages transactions at the packet level with minimal intervention from the processor, handles the Endpoints' FIFOs, monitors the status of the transactions and communicates with the AVR through a set of status and control registers.

The FIU is designed to operate in single-packet mode and to manage the USB packet protocol layer. To operate the FIU, the firmware must first enable the Endpoints of the FIU, and then select a direction and ping-pong capability. Once enabled, the Endpoints are in receive mode by default. The FIU notifies the processor when a valid token has been received. The data contained in the data packet will be supplied in the FIFO.

The processor transfers the data to and from the Host by interacting with each Endpoint's FIFO and Control and Status registers.

For example, when transmitting an IN packet, the FIU assembles the data of the Endpoint's FIFO in a USB packet, transmits the packet, and then signals the processor after the Host receives and acknowledges the packet. The FIU performs automatic data packet retransmission and DATA0/DATA1 PID toggling.

For SETUP tokens, the processor must parse the device request and then respond appropriately. After a SETUP token, there may be 0 or more DATA IN or DATA OUT packets for which the processor must either supply or receive the data.

6.7.8 USB Serial Interface Engine (SIE)

The SIE performs the following functions:

- NRZI data encoding/decoding
- Bit stuffing/un-stuffing
- CRC generation and checking
- ACKs and NACKs
- Identifying the type of a token
- Address checking
- Clock generation (via DPLL)

6.7.9 Control Transfers at Function EP0

Legend: DATA1/DATA0 = Data packet with DATA1 or DATA0 PID

DATA 1 (0) = Zero length DATA1 packet

Host	USB Macro	Microcontroller
SETUP Stage		
1. [SYNC]-[SETUP]		
2. [SYNC]-[DATA0]		
	3. Data are put in FIFO	
	4. If CRC OK, send:[SYNC]-[ACK]	
	5. if CRC OK, set RX_SETUP bit	
		6. INTERRUPT
		7. Read UISR (bit 0 is set)
		8. Read FCSR0 (RX_SETUP bit)
		9. Read FBYTE_CNT0
		10. Read FIFO 0
		11. Parse Data If Control Read Phase: Set Control Direction Clear RX_SETUP bit Fill FIFO with data Set TX_Packet_Ready If Control Write Phase: Clear Control Direction Clear RX_SETUP bit If No data Stage Phase: Clear Control Direction Set Data_End bit Set FORCE_STALL bit If Unsupported Command: Set FORCE_STALL bit
		12. SET UIAR (EP0 INTA)
Status Stage, No DATA Stage		
1. [SYNC]-[IN]		
	2. Send DATA1(0)	
3. If CRC Ok, send [SYNC]-[ACK]		
	4. Set TX_Complete bit	
		5. INTERRUPT
		6. Read UISR
		7. Read CSR

Host	USB Macro	Microcontroller
		8. If SET_ADDRESS Write Device Address to FADDR Set FADD Enable bit of Global State Registe If SET_CONFIGURATION with a 1: Set CONFIG bit of Global State Register
		9. Clear Tx_Complete bit
		10. Clear Data_End bit
		11. Set FORCE_STALL bit
		12. SET UIAR (EP0 INTA)
DATA Stage, Control READ		
1. [SYNC]-[IN]		
	2. If TX_Packet_Ready = 1 send DATA0 / DATA1 else, send STALL	
3. If CRC OK, send [SYNC]-[ACK]		
	4. Clear TX_Packet_Ready	
	5. Set TX_Complete bit	
		6. INTERRUPT
		7. Read UISR
		8. Read CSR
		9. Clear Tx_Complete bit
		10. If more data fill FIFO with data and set TX_Packet_Ready else if a NULL packet should be sent, set DATA_END and set TX_Packet_Ready else if all bytes sent and need to send a NULL packet, set DATA_END and set SET_FORCE_STALL
		11. SET UIAR (EP0 INTA)
STATUS/Early STATUS Stage with READ DATA Stage		
1. [SYNC]-[OUT]		

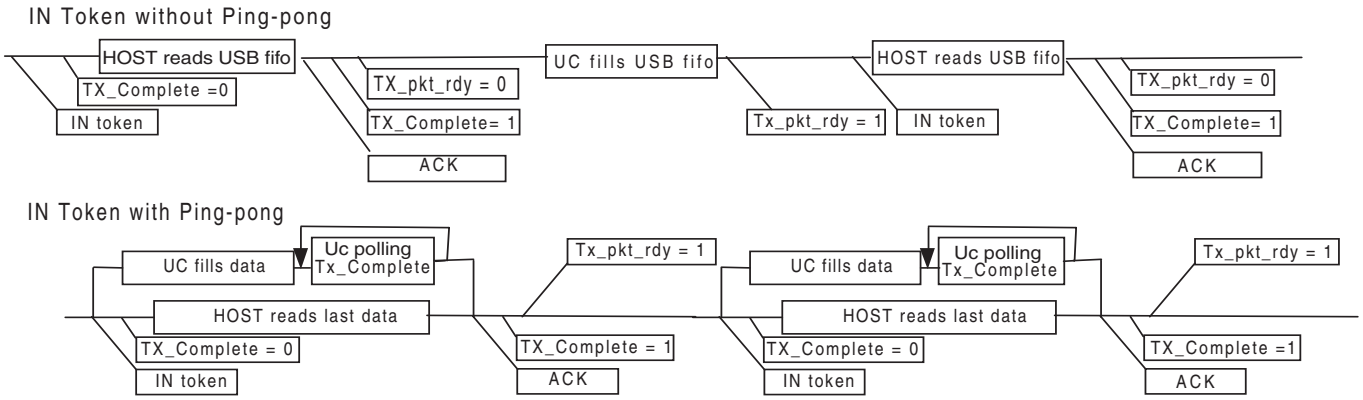
Host	USB Macro	Microcontroller
2. [SYNC]-[DATA1(0)]		
	3. If TX_Complete = 0 then send [SYNC]-[ACK] and set RX_OUT else send [SYNC]-[NACK]	
		4. INTERRUPT
		5. Read UISR
		6. Read CSR
		7. Clear RX-OUT
		8. Set Data_end
		9. Set Force_Stall Comment: A SETUP token will clear Data_End. Not cleared by firmware in case Host retries 1 through 3
	10.	11. SET UIAR (EP0 INTA)
DATA Stage, Control WRITE		
1. [SYNC]-[OUT]		
2. [SYNC]-[DATA1/DATA0]		
	3. Data are put in FIFO	
	4. If CRC ok send [SYNC]- [ACK]	
	5. if CRC ok, set RX_OUT	
		6. INTERRUPT
		7. Read UISR
		8. Read CSR
		9. Read FIFO
		10. Clear RX_OUT
		If last packet, Set Data_End Set Force_Stall
		11. SET UIAR (EP0 INTA)
STATUS Stage with WRITE DATA Stage		
1. [SYNC]-[IN]		
	2. Send DATA1(0)	
3. If CRC Ok, send [SYNC]-[ACK]		
	4. Set TX_Complete bit	
		5. INTERRUPT
		6. Read UISR

Host	USB Macro	Microcontroller
		7. Read CSR
		8. Clear Tx_Complete bit
		9. Clear Data_End bit
		10. Set FORCE_STALL bit
		11. SET UIAR (EP0 INTA)

6.7.10 Interrupt and Bulk In Transfers

The USB hardware automatically starts the Endpoint in the receive mode and NAKs all IN tokens, as long as the CSR[TX Packet Ready] is cleared. The processor checks this bit and if it is 0, it writes the data into the FIFO and then sets CSR[TX Packet Ready]. At the next IN token, the USB hardware sends the packet out and waits for an ACK. Until an ACK is received, the USB hardware will retransmit the packet. After receiving an ACK, the USB hardware clears the CSR[TX Packet Ready], signaling a successful completion to the processor.

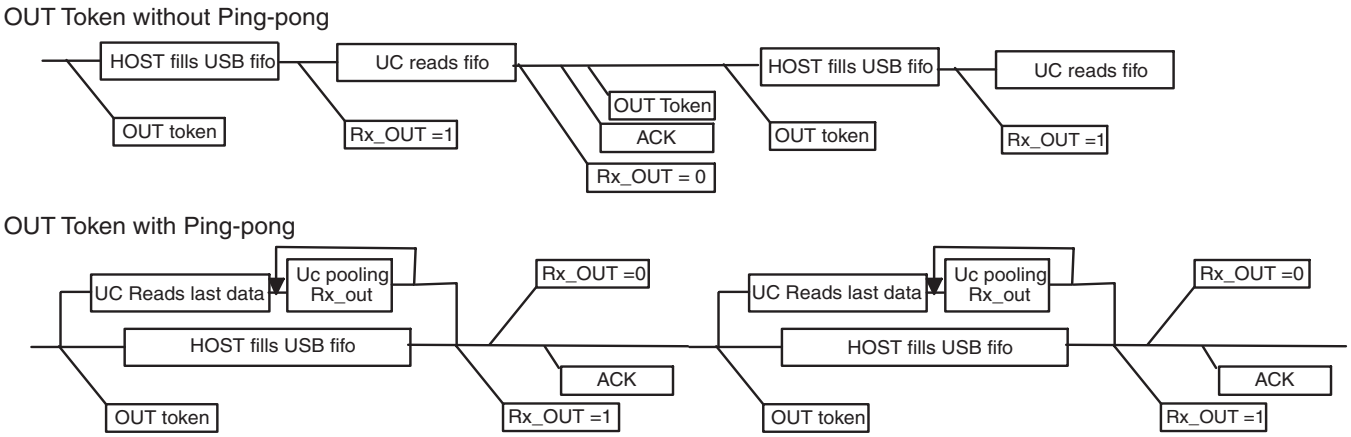
Figure 6-8. Bulk IN Transfers



6.7.11 Interrupt and Bulk OUT Transfers

The USB hardware automatically starts the Endpoint in receive mode. When an OUT token is received and if the CSR[RX OUT Packet] is cleared, it stores the data in the FIFO. It ACKs the Host if the data received are not corrupted and then interrupts the processor. If the CSR[RX OUT Packet] is set, the USB hardware responds with a NAK to the incoming OUT token. The processor checks the CSR[RX OUT Packet] and if it is 1, it reads the data from the FIFO and clears CSR[RX OUT Packet Ready].

Figure 6-9. Bulk OUT Transfers



6.7.12 Interrupt and Isochronous Transfers

Isochronous transfers use the same protocol with bulk transfers except that error correction and data packet retransmission are not supported:

- No ACK token
- No NAK token
- Data PID is always 0

6.7.13 Interrupt and Interrupt IN Transfers

Interrupt transfers use the same protocol with bulk IN transfers (Interrupt OUT is not supported in USB Spec 1.0).

6.7.14 Suspend Mode

A USB device enters the suspend mode only when requested by the USB Host through bus inactivity for at least 3 ms. The USB H/W detects this request, sets the SUSP bit of the Suspend/Resume register (SPRSR), and interrupts the processor if the interrupt is enabled. The processor should shut down any peripheral activity, enter power-down mode, and signal the USB H/W that it can enter the suspend mode by writing 1 to the sleep mode USB_Macro input pin. The firmware must then execute the SLEEP instruction and set the system to a power-down state. The oscillator, PLL, AVR, and all peripherals will stop and the SUSP pin will be set to high.The USB suspend interrupt hits twice; one time in the beginning of the suspend, and one time at the end of the suspend.

6.7.15 Resume Mode

Resume mode is signaled by a J-to-K state transition at the USB port. The USB H/W enables the oscillator/PLL and sets the RSM bit of the SPRSR, which generates an interrupt. The processor starts executing where it left off and services the interrupt. The firmware clears the RCVD RSM bit.

6.7.16 Remote Wakeup

During Remote Wakeup, a resume is signaled by the firmware by setting the RSM bit of the SPRSR register.The firmware must wait for some time (8 ms) before clearing the the RSM bit in order to give time to the Host to detect the resume signal on the bus.

6.8 UART0, UART1

UART0 and UART1 are 16550-compatible UARTs with some additional features.

The main features of the UARTs are:

- Can run 16550 software
- All registers are in 16550-compatible mode after reset
- Programmable baud rate generator
- Maximum data rate 921.6 Kbaud
- Exception handling using either prioritized interrupts or polled modes
- Parity, framing, and overrun error detection
- Two dedicated controller channels
- 16-byte transmit FIFO
- 16-byte plus 3 error bits receive FIFO
- 5-, 6-, 7-, and 8-bit word length
- Bi-directional handshaking modem control signals (available only for UART0)
- Line break generation and detection
- Multidrop mode: Address detection and generation
- Diagnostic loopback mode (with or without an echo)

6.8.1 Baud Rate Generator

The input to the baud rate generator is 14.769 MHz, which is derived from the internal clock generator.

6.8.2 Receiver

The UART detects the start of a received word by sampling the SIN signal until it detects a valid start bit. A low-level (that is, a space) on SIN is interpreted as a valid start bit if it is detected for more than 7 cycles of the sampling clock, which is 16 times the baud rate. Thereore, a space which is longer than 7/16 of the bit period is detected as a valid start bit. A space, which is 7/16 of a bit period or shorter, is ignored and the receiver continues to wait for a valid start bit. When a valid start bit has been detected, the receiver samples the SIN at the theoretical midpoint of each bit. It is assumed that each bit lasts 16 cycles of the sampling clock (1 bit period) so the sampling point is 8 cycles (0.5 bit periods) after the beginning of the bit. The first sampling point is sampled 24 cycles (1.5 bit periods) after the falling edge of the start bit was detected. Each subsequent bit is sampled 16 cycles (1 bit period) after the previous one.

6.8.3 Receive FIFO Operation

The 16-byte plus 3 error bits receive data FIFO is enabled by the FCR bit 0. The user can set the receiver trigger level.

6.8.4 Time-out

The receiver section includes a time-out mechanism in order to trace the time interval between the received words.

The RTO register contains the maximum bit periods, for which the UART will wait for the next word to arrive. Whenever the time-out counter expires (reaches $00), then a time-out indication interrupt will be issued.

The XR1[5] Start Time-out Control bit selects the Start Time-out and RTO load mechanism. If the XR1[5] bit is reset to 0 (16550 compatible mode), then the RTO loads the value 4 times the word length plus 12 on each LCR write operation. After reset, the word length is 5 bits and the RTO is $20. The time-out counter will then start counting down only in FIFO mode (FCR[0] is set) and if the Rx FIFO holds at least 1 word. If XR1[5] is set to 1, then the time-out function is available and in FIFO-disabled mode. The RTO value does not change with the LCR write operations. The time-out counter will start counting down whenever the RTO is not $00. In all cases, the core has immediate access to the contents of the RTO.

The time-out counter resets on a RHR read access from core or when a new word is completely received and transferred to the Receive Holding register or when the XR1[4] bit is forced to logic 1.

6.8.5 Receive Break

The break condition is detected by the receiver when the SIN line is held in the spacing state (logic 0) for longer than a full word transmission time (that is, the total time of all data, parity, and stop bits). At the moment of the low stop bit detection, the receiver asserts receive break indication (LSR[4]). The end of the receive break is detected by a high level for at least 2/16 of the bit period.

6.8.6 Transmitter

The start bit, data bits, parity bit, and stop bits are serially shifted, with the lowest significant bit first, on the falling edge of the UART clock. The LCR controls the number of data bits, the parity bit and the number of stop bits. When a word is written to THR (Transmit Holding register), it is transferred to the Shift register as soon as it is empty. When the transfer occurs, the THR ready (bit-5) in LSR is set until a new word is written to THR. If the Transmit Shift register and THR are both empty, the transmitter empty (bit-6) in LSR is set.

6.8.7 Transmit FIFO Operation

When the FCR[0] is set to logic 1 and XR2[7] is reset to 0, then the 16-byte transmit FIFO is enabled. The LCR[5] (and the XR2[5]) THR Empty bit indicates whether the Tx FIFO is empty or full. When XR2[4] is in logic 0 (16550 compatible mode), then the THR Empty bit indicates that Tx FIFO is empty. If XR2[4] is set to logic 1, then THR Empty bit indicates Tx FIFO is not full. If the THR Empty bit is active (high), then a THR Empty interrupt will be issued.

The firmware has two choices: 1) Reset XR2[4] bit to logic 0 and enable the THR Empty interrupt. On each THR Empty interrupt, the software will transfer up to 16 words to the THR (no check for FIFO full). 2) Set XR2[4] bit to logic 1 and enable THR Empty Interrupt. On each THR Empty interrupt, the software will know that it is possible to transfer more words to THR (until THR Empty bit is 0).

6.8.8 Time-guard

The Time-guard function allows the transmitter to insert an idle state between two words on the SOUT line. The duration of the idle state is programmed in U _TTG (Transmitter Time-guard). When this register is set to 0, no time-guard is generated.

6.8.9 Transmit Break

The transmitter can generate a break condition on the SOUT line when the Start Break command is issued by setting LCR[6] to logic 1. During the break condition SOUT line is held to space state (logic 0). The Start Break command will take effect right after the complete transmission of all words in Transmit Shift register and THR (or Tx FIFO, if enabled). No software

synchronization is needed. To remove the break condition on the SOUT line, a Stop Break command should be issued by resetting US_LCR[6] to logic 0. The UART generates minimum break duration of one word length. After the Stop Break commandhas been issued, the SOUT line returns to high level (idle state) for at least 12 bit periods to ensure that the end of break is correctly detected. The transmitter resumes the normal operation.

6.8.10 Interrupt Generation

The UART prioritizes interrupts into five levels and records these in the Interrupt Identification register (IIR). The five levels of interrupt conditions in order of priority are: Receiver Line Status, RHR Ready, Time-out, THR Ready, and Modem Status. When the CPU accesses the IIR, the UART freezes the contents of IIR and indicates the highest priority pending an interrupt to the CPU. During this CPU access, the UART records new interrupts, but does not change its current indication until the access is complete.

The Interrupt Enable register (IER) controls which interrupt condition will issue an interrupt to the core. Disabling all interrupts, the UART works in polled mode.

6.8.11 Handshaking Signals

Both UARTs implement all the modem control handshaking signals, but none of those signals are mapped to any I/O port for UART1. For UART0, the nOUT1 and nOUT2 signals are not mapped to any I/O ports. The only available handshaking signals are the nRTS, nCTS, nDSR, nDTR, nCD, and nRI of UART0.

All modem control signals are bi-directional and the firmware has the full control of their status.

6.8.12 Loopback Mode

In the diagnostic loopback mode, the UART deactivates the output signals (stack at logic 1) and loops-back their status to the proper input signals. The UART offers the option of not stacking the UART output pins at logic 1 but rather echoing their status at the external UART interface.

The loopback mode cannot be asserted if the direction of the modem control signals (MDR) are not in the default state.

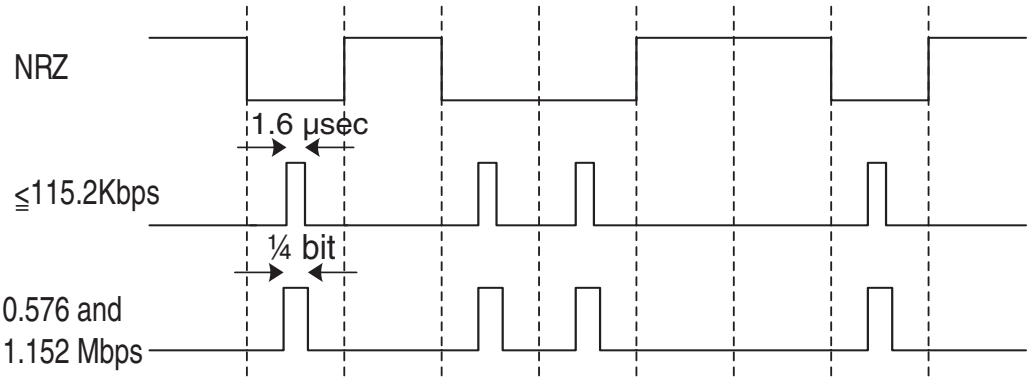
6.9 IrDA 1.0 Codec

The IrDA 1.0 codec can be attached on either the UART0 or UART1. The operation of IrDA codec is controlled by the “IrDA Control Register (IRDACR)” on page 53.

The IrDA 1.0 codec can effect the UART’s SIN and SOUT signals by applying 3/16 or 4/16 modulation of the Return-to-Zero encoding scheme that the UART produces.

According to the IrDA standard, for data rates up to and including 1.152 Mbit/s, the RZI modulation scheme is used and a 0 is represented by a light pulse. For rates up to and including 115.2 Kbit/s, the optical pulse duration is nominally 3/16 of a bit duration (or 3/16 of a 115.2 Kbit/s bit duration). For 0.576 Mbit/s and 1.152 Mbit/s, the optical pulse duration is nominally 4/16 of a bit duration.

Figure 6-10. Pulse Duration



6.10 Watchdog Timer

The main features of the Watchdog Timer (WDT) are:

- 3 MHz clock
- 22-bit up-counter
- Programmable prescaler
- Write access protection on the timer disable

The WDT is used to prevent a system lock-up if the software becomes trapped in a deadlock.

The WDT is clocked with a 3 MHz clock from the clock generator. In a normal operation, the user resets the WDT at regular intervals, using the WDR instruction. By controlling the WDT prescaler, the Watchdog reset interval can be adjusted (see WDTCR register, WDP2..0 bits). Eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the system resets and executes from the reset vector.

To prevent unintentional disabling of the watchdog, a special turnoff sequence must be followed when the WDT is disabled. Refer to the description of the WDTCR register for details.

6.11 Serial Peripheral Interface (SPI)

The AT76C713 SPI features are:

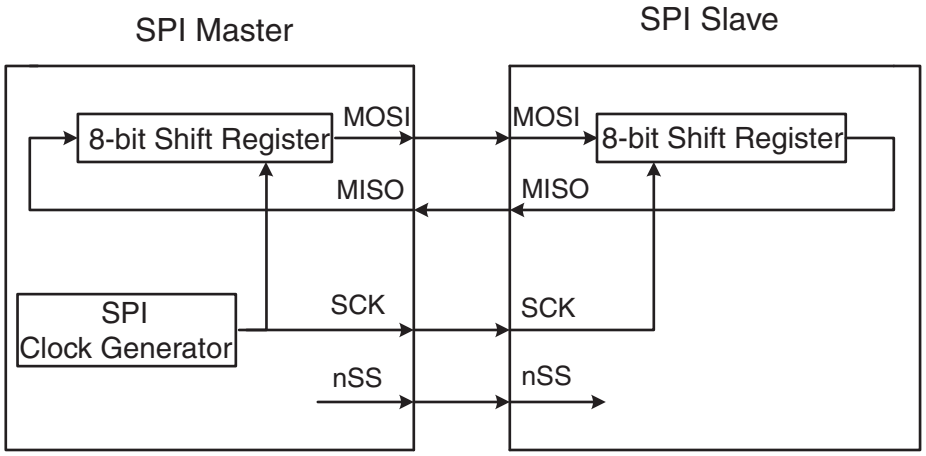
- Full-duplex 3-wire synchronous data transfers
- Master or slave operation
- LSB first or MSB first data transfer
- Four programmable bit rates
- End-of-Transmission interrupt request
- Write collision flag protection
- Wake-up from idle or power-down mode (slave mode only)

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfers between the AT76C713 and peripheral devices (memories, controllers, etc.). The interconnection between master and slave SPI controllers is shown in Figure 6-11.

The two shift registers of the master and the slave SPI controllers can be considered as one distributed 16-bit circular shift register. As Figure 6-11 shows, when data is shifted from the master

to the slave, data is also shifted in the opposite direction simultaneously. This means that during one shift cycle the data in the master and the slave are interchanged.

Figure 6-11. Interconnection Between Master and Slave SPI Controllers



The PB4(SCK) pin is the clock output in the master mode and the clock input in the slave mode. Writing to the SPI Data register of the SPI master starts the SPI clock generator and the data written shifts out of the Master MOSI pin and into the Slave MOSI pin. After shifting one byte, the SPI clock generator stops setting the End-of-Transmission flag (SPIF). If the SPI interrupt enable bit (SPIE) in the SPCR register is set, an interrupt is issued. The Slave Select input (nSS) is set low to select an individual slave SPI device.

The system is single-buffered in both transmit and receive directions. This means that bytes to be transmitted cannot be written to the SPI Data register before the entire shift cycle is completed. When receiving data, we must read the received byte before the next reception is started. Otherwise, the first byte is lost.

When the SPI is enabled (PERIPHEN register, SPI bit), the data direction of the MOSI, MISO, SCK, and nSS pins is overridden, according to Table 6-14.

Table 6-14. Data Direction of SPI pins

Pin	SPI Master	SPI Slave	Description
PB4	User defined	Input	nSS (SPI Slave Select Input)
PB5	Output	Input	MOSI (SPI Bus Master Output/Slave Input)
PB6	Input	Output/Hi-Z	MISO (SPI Bus Master Input/Slave Output)
PB7	Output	Input	SCK (SPI Bus Serial Clock)

6.11.1 nSS(PB4) Pin Functionality

When the SPI is configured as a master (MSTR in SPCR register), the user can determine the direction of the nSS (PB4) pin. If the nSS is configured as an output, the pin is a general output pin, which does not affect the SPI system. If the nSS is configured as an input, it must be held high to ensure master SPI operation. If the nSS pin is driven low by peripheral circuitry when the SPI is configured as master with the nSS pin defined as an input, then the SPI system interprets this as another master selecting the SPI as a slave and starts to send data to it. To avoid bus contention, the SPI system takes the following actions

1. The MSTR bit in SPCR is cleared and the SPI system becomes a slave. As a result of the SPI becoming a slave, the MOSI and SCK pins become inputs.
2. The SPIF flag in SPSR is set, and if the SPI interrupt is enabled and the I-bit in SREG is set, then the interrupt routine will be executed.

Thus, when the interrupt-driven SPI transmittal is used in the master mode and there exists a possibility that the nSS is driven low, the interrupt should always check that the MSTR bit is still set. Once the MSTR bit has been cleared by a slave select, it must be set by the user to re-enable the SPI master mode.

When the SPI is configured as a slave, the nSS pin is always an input. When nSS is held low, the SPI is activated and MISO becomes an output. All other pins (MOSI and SCK) are inputs. When nSS is driven high, all pins are inputs and the SPI is passive, which means that it will not receive incoming data. Note that the SPI logic will be reset once the nSS pin is brought high. If the nSS pin is brought high during a transmission, the SPI will stop sending and receiving immediately, and both data received and data sent must be considered lost.

6.11.2 SPI Modes

The SPI bus protocol includes 4 modes. These modes determine the relationship between the serial clock and the data bits. The SPI mode is fully programmable using the SPI Control register (SPICR). The four modes are determined with the CPOL and CPHA bits, as illustrated in the following four figures.

Note: In all modes, the data pins MOSI and MISO are driven in the opposite SCLK edge. For example, if the data input is latched on the rising edge, then the data output is driven on the falling edge.

Figure 6-12. SPI Mode 0

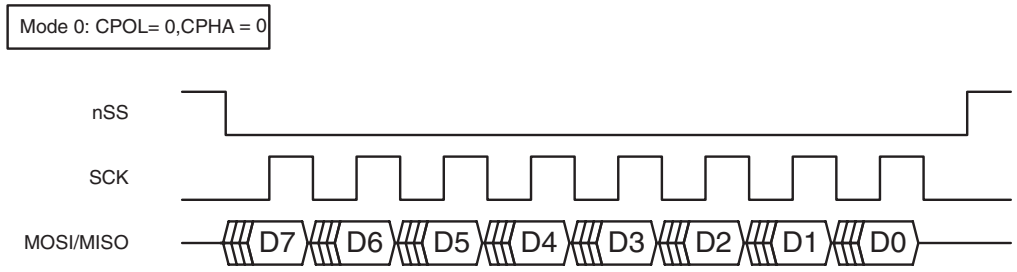


Figure 6-13. SPI Mode 1

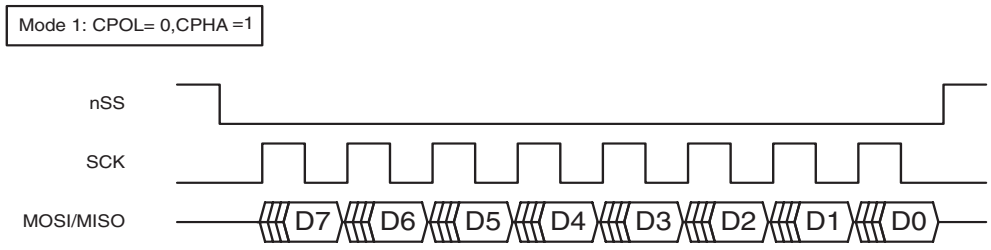


Figure 6-14. SPI Mode 2

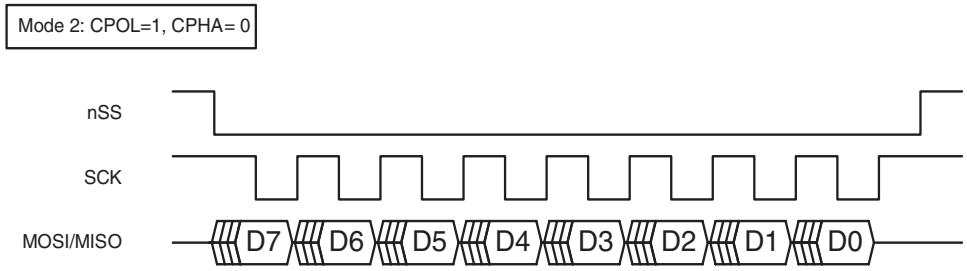
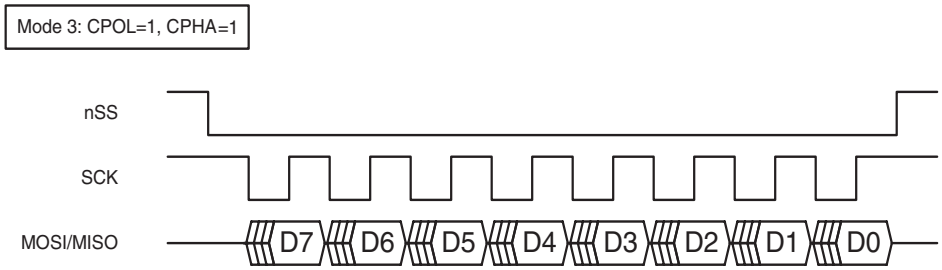


Figure 6-15. SPI Mode 3



6.12 JTAG Interface and On-chip Debug System

The main features of the JTAG and On-chip Debug (OCD) System are:

- JTAG (IEEE Std. 1149.1 Compliant) interface
- Boundary-scan capabilities according to the IEEE Std. 1149.1 (JTAG)
- Debugger access to:
  - All Internal Peripheral Units
  - Internal and external data RAM
  - Program memory
  - Internal register file
  - Program counter
- Extensive on-chip debug support for break conditions, including:
  - AVR break instruction
  - Break on change of program memory flow
  - Single step break
  - Program memory break points on a single address or address range
  - Data memory break points on a single address or address range

The AVR IEEE Std. 1149.1 compliant JTAG interface can be used for the following:

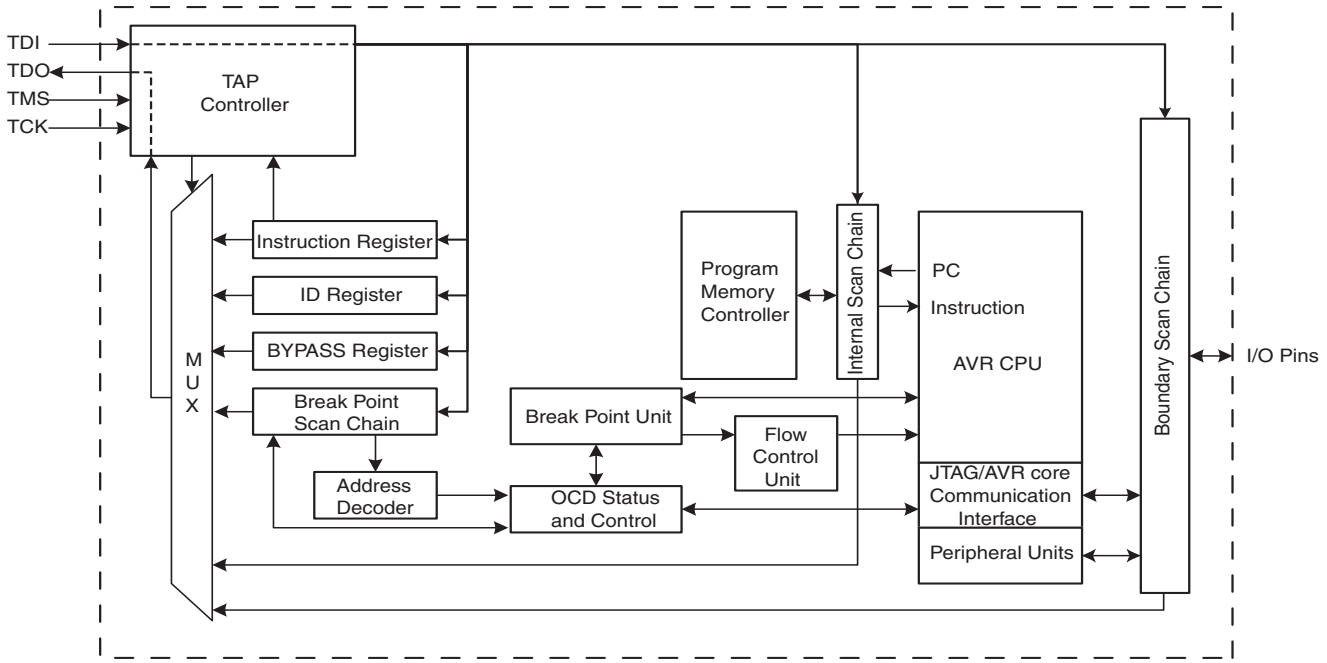
- Testing PCBs by using the JTAG boundary-scan capability
- On-chip debugging

Detailed descriptions for the boundary-scan chain can be found in the section “IEEE 1149.1 (JTAG) Boundary-scan” of the IEEE 1149.1 Standard. The on-chip debug support contains confidential JTAG instructions and is distributed within Atmel and to selected third party vendors only.

Figure 6-16 shows a block diagram of the JTAG interface and the on-chip debug system. The TAP controller selects either the JTAG Instruction register or one of several Data registers as the scan chain (Shift register) between the TDI input and TDO output. The Instruction register holds JTAG instructions controlling the behavior of a Data register.

The ID register, Bypass register, and the boundary-scan chain are the Data registers used for board-level testing. The internal scan-chain and break point scan chain are used for on-chip debugging purposes only.

Figure 6-16. JTAG Interface and On-chip Debug System Block Diagram



6.12.1 Test Access Port (TAP)

The JTAG interface is accessed through four pins. In JTAG terminology, these pins constitute the Test Access Port (TAP). These four pins are:

- TMS: Test Mode Select - This pin is used for navigating through the TAP-controller state machine
- TCK: Test Clock - JTAG operation is synchronous to TCK
- TDI: Test Data In - Serial input data to be shifted in to the Instruction register or Data register (scan chains)
- TDO: Test Data Out - Serial output data from Instruction register or Data register

The IEEE Std. 1149.1 also specifies the optional TAP signal TRST (Test ReSeT) which is not provided.

For the on-chip debug system, in addition to the JTAG interface pins, the nRST pin is monitored by the debugger in order to detect external reset sources. The debugger can also pull the nRST pin low (using the AVR_RESET command) in order to reset the entire system.

6.12.2 TAP Controller

The TAP controller is a 16-state finite-state machine that controls the operation of the boundary-scan circuitry or the on-chip debug system. The state transitions depicted in Figure 6-17 depend

on the signal present on the TMS (shown adjacent to each state transition) at the time of rising edge at the TCK. The initial state after a Power-on Reset is a Test-Logic-Reset.

As a definition in this document, the LSB is shifted in and out first for all Shift registers.

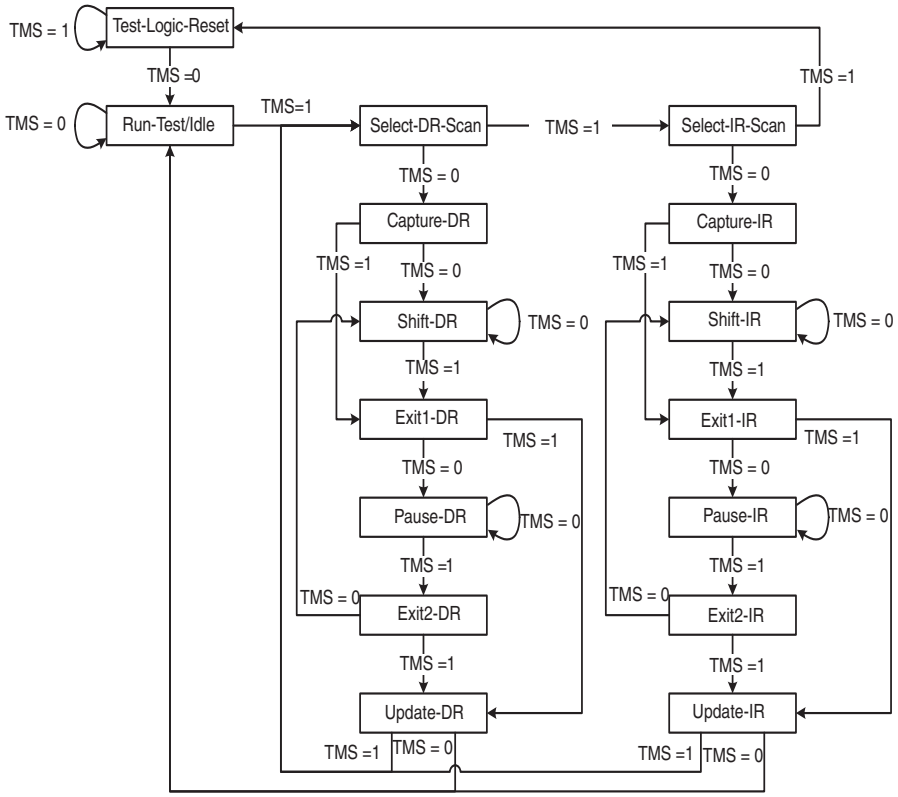
Assuming Run-Test/Idle is the present state, a typical scenario for using the JTAG interface is as follows:

- In the TMS input, apply the sequence 1, 1, 0, 0 at the rising edges of the TCK to enter the Shift Instruction register (Shift-IR) state. While in this state, shift the four bits of the JTAG instructions into the JTAG Instruction register from the TDI input at the rising edge of TCK. The TMS input must be held low during input of the 3 LSBs in order to remain in the Shift-IR state. The MSB of the instruction is shifted in when this state is left by setting the TMS to high. While the instruction is shifted in from the TDI pin, the captured IR-state 0x01 is shifted out on the TDO pin. The JTAG instruction selects a particular Data register as path between TDI and TDO and controls the circuitry surrounding the selected Data register.
- Apply the TMS sequence 1, 1, 0 to re-enter the Run-Test/Idle state. The instruction is latched onto the parallel output from the Shift register path in the Update-IR state. The Exit-IR, Pause-IR, and Exit2-IR states are only used for navigating the state machine.
- At the TMS input, apply the sequence 1, 0, 0 at the rising edges of the TCK to enter the Shift Data register (Shift-DR) state. While in this state, upload the selected Data register (selected by the present JTAG instruction in the JTAG Instruction register) from the TDI input at the rising edge of the TCK. In order to remain in the Shift-DR state, the TMS input must be held low during the input of all bits except the MSB. The MSB of the data is shifted in when this state is left by setting the TMS high. While the Data register is shifted in from the TDI pin, the parallel inputs to the Data register captured in the Capture-DR state is shifted out on the TDO pin.
- Apply the TMS sequence 1, 1, 0 to re-enter the Run-Test/Idle state. If the selected Data Register has a latched parallel-output, the latching takes place in the Update-DR state. The Exit-DR, Pause-DR, and Exit2-DR states are only used for navigating the state machine.

As shown in the state diagram, the Run-Test/Idle state need not be entered between selecting JTAG instruction and using the Data registers. Some JTAG instructions may select certain functions to be performed in the Run-Test/Idle, making it unsuitable as an idle state.

Note: Independent of the initial state of the TAP controller, the Test-Logic-Reset state can always be entered by holding the TMS high for five TCK clock periods.

Figure 6-17. JTAG TAP State Diagram



6.12.3 Using the On-chip Debug System

As shown in Figure 6-16 on page 35, the hardware support for the on-chip debugging consists mainly of the following:

- A scan chain on the interface between the internal AVR CPU and the internal peripheral units
- Break point unit
- Communication interface between the CPU and JTAG system

All read or modify/write operations needed for implementing the debugger are done by applying AVR instructions via the internal AVR CPU scan chain. The CPU sends the result to an I/O memory mapped location which is part of the communication interface between the CPU and the JTAG system.

The break point unit implements a break on the change of program flow, single step break, two program memory break points, and two combined break points. Together, the four break points can be configured as either:

- Four single program memory break points
- Three single program memory break point plus 1 single data memory break point
- Two single program memory break points plus 2 single data memory break points
- Two single program memory break points plus1 program memory break point with mask (“range break point”)
- Two single program memory break points plus 1 data memory break point with mask (“range break point”)

However, a debugger may use one or more of these resources for its internal purpose, leaving less flexibility to the end user.

6.12.4 On-chip Debug Specific JTAG Instructions

The on-chip debug support contains private JTAG instructions and distributed within Atmel and to selected third party vendors only. Instruction opcodes are listed for reference in the Table 6-15.

Table 6-15. OCD Specific JTAG Instructions

Opcode	Description
PRIVATE 0; $8	Private JTAG instruction for accessing On-chip debug system
PRIVATE 1; $9	Private JTAG instruction for accessing On-chip debug system
PRIVATE 2; $A	Private JTAG instruction for accessing On-chip debug system
PRIVATE 3; $B	Private JTAG instruction for accessing On-chip debug system

6.13 IEEE 1149.1 (JTAG) Boundary-scan

Features:

- JTAG (IEEE Std. 1149.1 compliant) interface
- Boundary-scan capabilities according to the JTAG standard
- Full scan of all port functions, as well as analog circuitry having off-chip connections
- Supports the optional IDCODE instruction
- Additional public AVR_RESET instruction to reset the AVR

The boundary-scan chain has the capability of driving and observing the logic levels on the digital I/O pins. At system level, all ICs having JTAG capabilities are connected serilly by the TDI/TDO signals to form a long Shift register. An external controller sets up the devices to drive values at their output pins and observe the input values received from other devices. The controller compares the received data with the expected result. In this way, the boundary-scan provides a mechanism for testing interconnections and integrity of components on printed circuits boards by using the four TAP signals only.

The four IEEE 1149.1 defined mandatory JTAG instructions IDCODE, BYPASS, SAMPLE/PRELOAD, and EXTEST, as well as the AVR-specific public JTAG instruction AVR_RESET can be used for testing the printed circuit board. Initial scanning of the Data register path will show the ID code of the device, since IDCODE is the default JTAG instruction. It may be desirable to have the AVR device in reset during test mode. If not reset, inputs to the device may be determined by the scan operations, and the internal software may be in an undetermined state when exiting the test mode. Entering reset, the outputs of any port pin will instantly enter the high impedance state, making the HIGHZ instruction redundant. If needed, the BYPASS instruction can be issued to make the shortest possible scan chain through the device. The device can be set in the reset state either by pulling the external nRST pin low, or issuing the AVR_RESET instruction with appropriate setting of the Reset Data register.

The EXTEST instruction is used for sampling external pins and loading output pins with data. The data from the output latch will be driven out on the pins as soon as the EXTEST instruction is loaded into the JTAG IR register. The SAMPLE/PRELOAD instruction should also be used for setting initial values to the scan ring in order to avoid damaging the board when issuing the

EXTEST instruction for the first time. The SAMPLE/PRELOAD instruction can also be used for taking a snapshot of the external pins during normal operation of the part.

When using the JTAG interface for the boundary-scan, using a JTAG TCK clock frequency higher than the internal chip frequency is possible. The chip clock is not required to run.

6.13.1 Data Registers

The Data registers relevant to the boundary-scan operations include:

- Bypass register
- Device Identification register
- Reset register
- Boundary-scan chain

6.13.1.1 Bypass Register

The Bypass register consists of a single Shift register stage. When the Bypass register is selected as a path between the TDI and TDO, the register is reset to 0 when leaving the Capture-DR controller state. The Bypass register can be used to shorten the scan chain on a system when the other devices are to be tested.

6.13.1.2 Device Identification Register

Bit	Field	Value	**Description**
31:28	Version	$1	Version is a 4-bit number identifying the revision of the component
27:12	Part number	$C712	The part number is a 16-bit code identifying the component. The unique JTAG Part Number for AT76C713 and AT76C713 is $C712
11:1	Manufacturer ID	$01F	The Manufacturer ID is an 11-bit code identifying the manufacturer. The JTAG manufacturer ID for ATMEL is $01F
0	1	$1	

6.13.1.3 Reset Register

The Reset register is a test Data register used to reset the part. Since the AVR tri-states Port Pins when reset, the Reset register can also replace the function of the not implemented optional JTAG instruction HIGHZ.

A high value in the Reset register corresponds to pulling the external Reset low. The part is reset, as long as there is a high value present in the Reset register. The output from this Data register is not latched, so the reset will take place immediately.

6.13.1.4 Boundary-scan Chain Register

The boundary-scan chain has the capability of driving and observing the logic levels on the digital I/O pins. See the “Boundary-scan Chain” on page 41 for a complete description.

6.13.2 Boundary-scan Specific JTAG Instructions

The Instruction register is 4-bit wide and supports up to 16 instructions. Listed below are the JTAG instructions useful for the boundary-scan operation. Note that the optional HIGHZ instruc-

tion is not implemented, but all outputs with tri-state capability can be set in a high-impedance state by using the AVR_RESET instruction, since the initial state for all port pins is tri-state.

As a definition in this data sheet, the LSB is shifted in and out first for all Shift registers.

The OPCODE for each instruction is shown behind the instruction name in a hex format. The text describes which Data register is selected as the path between the TDI and TDO for each instruction.

6.13.2.1 EXTEST; $0

EXTEST; $0 is the mandatory JTAG instruction for selecting the boundary-scan chain as the Data register for testing circuitry external to the AVR package. The contents of the latched outputs of the boundary-scan chain are driven out as soon as the JTAG IR-register is loaded with the EXTEST instruction.

The active states are:

- Capture-DR: Data on the external pins is sampled into the boundary-scan chain
- Shift-DR: The internal scan chain is shifted by the TCK input
- Update-DR: Data from the scan chain is applied to output pins

6.13.2.2 IDCODE; $1

IDCODE; $1 is an optional JTAG instruction selecting the 32 bit ID-register as the Data register. The ID-register consists of a version number, a device number, and the manufacturer code chosen by JEDEC. This is the default instruction after power-up. The active states are:

- Capture-DR: Data in the IDCODE register is sampled into the boundary-scan chain
- Shift-DR: The IDCODE scan chain is shifted by the TCK input.

6.13.2.3 SAMPLE_PRELOAD; $2

SAMPLE_PRELOAD; $2 is a mandatory JTAG instruction for pre-loading the output latches and taking a snap-shot of the input/output pins without affecting the system operation. However, the output latches are not connected to the pins. The boundary-scan chain is selected as Data Register. The active states are:

- Capture-DR: Data on the external pins are sampled into the boundary-scan chain.
- Shift-DR: The boundary-scan chain is shifted by the TCK input
- Update-DR: Data from the boundary-scan chain is applied to the output latches. However, the output latches are not connected to the pins

6.13.2.4 AVR_RESET; $C

AVR_RESET; $*C* is the AVR-specific public JTAG instruction for forcing the AVR device into the reset mode or releasing the JTAG reset source. The TAP controller is not reset by this instruction. The one bit Reset register is selected as the Data register. Note that the reset will be active as long as there is a logic 1 in the reset chain. The output from this chain is not latched. The active states are:

- Shift-DR: The Reset register is shifted by the TCK input

6.13.2.5 *BYPASS; $F*

BYPASS; $F is the mandatory JTAG instruction selecting the Bypass register for Data register. The active states are:

- Capture-DR: Loads a logic 0 into the Bypass register
- Shift-DR: The Bypass register cell between TDI and TDO is shifted

6.13.3 Boundary-scan Chain

The boundary-scan chain consists of two kind of cells. The first one is a standard scan cell with observe and drive capabilities, while the second one is an observe-only cell.

6.13.3.1 *AT76C713 Boundary-scan Order*

Table 6-16 shows the scan order between the TDI and TDO when the boundary-scan chain is selected as the data path. Bit 0 is the LSB; the first bit scanned in, and the first bit is scanned out. The scan order follows the pin-out order. Therefore, the bits of Port A for example are not scanned in order.

Figure 6-18 shows an I/O pin with scan logic.

Figure 6-18. I/O Pin With Scan Logic

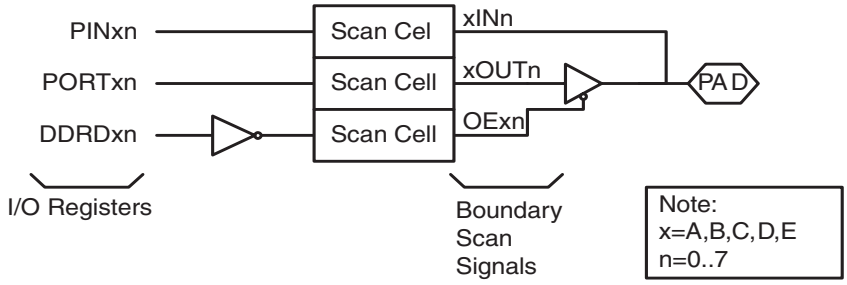


Table 6-16. AT76C713 Boundary-scan Order

Bit Number	Signal Name	Description
(LSB) 0	TEST_EN	(observe only)
1	TEST_CK	(observe only)
2	nRST	(observe only)
3	EOUT7	PE7
4	EIN7	
5	EOE7	
6	EOUT6	PE6
7	EIN6	
8	EOE6	
9	EOUT5	PE5
10	EIN5	
11	EOE5	

Table 6-16. AT76C713 Boundary-scan Order (Continued)

Bit Number	Signal Name	Description
12	EOUT4	PE4
13	EIN4	
14	EOE4	
15	EOUT3	PE3
16	EIN3	
17	EOE3	
18	EOUT2	PE2
19	EIN2	
20	EOE2	
21	EOUT1	PE1
22	EIN1	
23	EOE1	
24	EOUT0	PE0
25	EIN0	
26	EOE0	
27	PMODE0	
28	PMODE1	
29	COUT7	PC7
30	CIN7	
31	COE7	
32	COUT6	PC6
33	CIN6	
34	COE6	
35	COUT5	PC5
36	CIN5	
37	COE5	
38	COUT4	PC4
39	CIN4	
40	COE4	
41	COUT3	PC3
42	CIN3	
43	COE3	
44	COUT2	PC2
45	CIN2	
46	COE2	

Table 6-16. AT76C713 Boundary-scan Order (Continued)

Bit Number	Signal Name	Description
47	COUT1	PC1
48	CIN1	
49	COE1	
50	COUT0	PC0
51	CIN0	
52	COE0	
53	BOUT7	PB7
54	BIN7	
55	BOE7	
56	BOUT6	PB6
57	BIN6	
58	BOE6	
59	BOUT5	PB5
60	BIN5	
61	BOE5	
62	BOUT4	PB4
63	BIN4	
64	BOE4	
65	BOUT3	PB3
66	BIN3	
67	BOE3	
68	BOUT2	PB2
69	BIN2	
70	BOE2	
71	DOUT7	PD7
72	DIN7	
73	DOE7	
74	DOUT6	PD6
75	DIN6	
76	DOE6	
77	DOUT5	PD5
78	DIN5	
79	DOE5	
80	DOUT4	PD4
81	DIN4	
82	DOE4	

Table 6-16. AT76C713 Boundary-scan Order (Continued)

Bit Number	Signal Name	Description
83	DOUT3	PD3
84	DIN3	
85	DOE3	
86	DOUT2	PD2
87	DIN2	
88	DOE2	
89	DOUT1	PD1
90	DIN1	
91	DOE1	
92	DOUT0	PD0
93	DIN0	
94	DOE0	
95	SUSPEND	
96	AOUT7	PA7
97	AIN7	
98	AOE7	
99	AOUT6	PA6
100	AIN6	
101	AOE6	
102	AOUT5	PA5
103	AIN5	
104	AOE5	
105	AOUT4	PA4
106	AIN4	
107	AOE4	
108	AOUT3	PA3
109	AIN3	
110	AOE3	
111	USB_ATTACH	
112	nCS1	
113	nCS0	
114	nFWR	
115	nFRD	
116	AOUT2	PA2
117	AIN2	
118	AOE2	

Table 6-16. AT76C713 Boundary-scan Order (Continued)

Bit Number	Signal Name	Description
119	AOUT1	PA1
120	AIN1	
121	AOE1	
122	AOUT0	PA0
123	AIN0	
124	AOE0	
125	BOUT1	PB1
126	BIN1	
127	BOE1	
128	BOUT0	PB0
129	BIN0	
130	BOE0	
131	Idle-mode (sleep state)	(observe only)
132	Power-down (sleep state)	(observe only)
133	AVR clock stopped	(observe only)
134	PLL Stable Indication	(observe only)
(MSB) 135	PLL lock signal	(observe only)

7. I/O Space (Register Description)

The I/O space definition of the AT76C713 is shown in Table 7-1. This space is defined in the area $00 - $3F and can be directly accessed by IN and OUT instructions or by ordinary SRAM accesses in the area $20-$5F. The notation used (with the SRAM address in parentheses), will be followed in the rest of this document.

Table 7-1. AT76C713 I/O Space

I/O Address (SRAM Address)	Name	Function
$3F($5F)	SREG	Status register
$3E($5E)	SPH	Stack pointer high
$3D($5D)	SPL	Stack pointer low
$3C($5C)	IDR	OCD Debug register
$39($59)	TIMSK	Timer Interrupt Mask register
$38($58)	TIFR	Timer Interrupt Flag register
$37($57)	EIMSK	External Interrupt Mask register
$35($55)	MCUCR	MCU General Control register
$34($54)	MCUSR	MCU Status register
$33($53)	TCCR0	Timer0 Control registerregister

Table 7-1. AT76C713 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$32($52)	TCNT0	Timer0 (8-bit)
$31($51)	PRELD0	Pre-load register 0
$2E($4E)	TCCR1B	Timer1 Control register B
$2D($4D)	TCNT1H	Timer1 high byte
$2C($4C)	TCNT1L	Timer1 low byte
$2B($4B)	OCR1AH	Timer1 Output Compare register A High Byte
$2A($4A)	OCR1AL	Timer1 Output Compare register a low byte
$29($49)	OCR1BH	Timer1 Output Compare register B high byte
$28($48)	OCR1BL	Timer1 Output Compare register B low byte
$27($47)	ICR1H	Timer1 Input Capture register high byte
$26($46)	ICR1L	Timer1 Input Capture registerregister low byte
$25($45)	TCCR2	Timer2 Control register
$24($44)	TCNT2	Timer2 (8-bit)
$23($43)	PRELD2	Pre-load register 2
$22($42)	IRDACR	IrDA Control register
$21($41)	WDTCR	Watchdog Timer Control register
$1F($3F)	PMOD	Program Mode (PMODE0, PMODE1 pins value)
$1D($3D)	EMICRB	External Memory Interf. Control registerB
$1C($3C)	EMICRA	External Memory Interf. Control register A
$1B($3B)	PORTA	Data register, Port A
$1A($3A)	DDRA	Data Direction register, Port A
$19($39)	PINA	Input pins, Port A
$18($38)	PORTB	Data register, Port B
$17($37)	DDRB	Data Direction register, Port B
$16($36)	PINB	Input Pins, Port B
$15($35)	PORTC	Data register, Port C
$14($34)	DDRC	Data Direction register, Port C
$13($33)	PINC	Input Pins, Port C
$12($32)	PORTD	Data register, Port D
$11($31)	DDRD	Data Direction register, Port D
$10($30)	PIND	Input Pins, Port D
$0F($2F)	SPDR	SPI I/O Data register
$0E($2E)	SPSR	SPI Status register
$0D($2F)	SPCR	SPI Control register
$0C($2C)	CLK_CNTR	Clock Control register

Table 7-1. AT76C713 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$0B($2B)	PERIPHEN	Peripheral Enable register
$0A($2A)	PORTE	Data register, Port E
$09($29)	DDRE	Data Direction register, Port E
$08($28)	PINE	Input pins, Port E

AVR Status Register (SREG)

addr $3F($5F) 8 bits

Bit	Field	AVR	Description
7	I:Global interrupt enable	R/W	When set, the interrupts are enabled. The individual interrupt enable control is performed in the Individual Mask registers. This bit is cleared by H/W after an interrupt has occurred and is set by the RETI instruction to enable subsequent interrupts
6	T: Bit copy storage	R/W	Bit load (BLD) and bit store (BST) instructions use the T bit as source and destination for the operated bit
5	H: Half carry flag	R/W	Indicates a half carry in some arithmetic operations
4	S: Sign bit	R/W	Is an eXclusive OR between the negative flag N and the two's complement overflow flag V
3	V: Two's complement overflow flag	R/W	Supports two's complement arithmetic
2	N: Negative flag	R/W	When set, this indicates a negative result in arithmetic and logic operations
1	Z: Zero flag	R/W	When set, this indicates a 0 result after the different arithmetic and logic operations
0	C: Carry flag	R/W	When set, this indicates a carry in the arithmetic or logic operations

Stack Pointer (SP)

addr $3E($5E), $3D($5D) 11 bits

Bit	Field	AVR	Description
15:11	-	R/W	SPH
10	SP10	R/W	
9	SP9	R/W	
8	SP8	R/W	

Bit	Field	AVR	Description
7	SP7	R/W	SPL
6	SP6	R/W	
5	SP5	R/W	
4	SP4	R/W	
3	SP3	R/W	
2	SP2	R/W	
1	SP1	R/W	
0	SP0	R/W	

External Memory Interface Control Register A (EMICRA)

addr $1C($3C) 8 bits

Bit	Field	Default	AVR	Description
7	RW1	0	R/W	These bits control the Wait states inserted in the corresponding (read, write and ALE) signals
6	RW0	0	R/W	
5	RM1	0	R/W	These bits control the mode (waveform) of the corresponding (read, write and ALE) signals
4	RM0	0	R/W	
3	WW1	0	R/W	These bits control the Wait states inserted in the corresponding (read, write and ALE) signals
2	WW0	0	R/W	
1	WM1	0	R/W	These bits control the mode (waveform) of the corresponding (read, write and ALE) signals
0	WM0	0	R/W	

External Memory Interface Control Register B (EMICRB)

addr $1D ($3D) 8 bits

Bit	Field	Default	AVR	Description
7	AW1	0	R/W	These bits control the Wait states inserted in the corresponding (read, write and ALE) signals
6	AW0	0	R/W	
5	AM1	0	R/W	Thse bits control the mode (waveform) of the corresponding (read, write and ALE) signals
4	AM0	0	R/W	

<table>
<tr><th>Bit</th><th>Field</th><th>Default</th><th>AVR</th><th>Description</th></tr>
<tr><td>3:2</td><td>—</td><td>0</td><td>R/W</td><td></td></tr>
<tr><td>1</td><td>EMD0</td><td>0</td><td>R/W</td><td rowspan="2">These bits select the external memory interface mode according to the following table
<table><tr><th>EMD1</th><th>EMD0</th><th>External Memory Interface Mode</th></tr><tr><td>0</td><td>0</td><td>FIFO</td></tr><tr><td>0</td><td>1</td><td>Reserved</td></tr><tr><td>1</td><td>0</td><td>Demultiplexed</td></tr><tr><td>1</td><td>1</td><td>Multiplexed</td></tr></table></td></tr>
<tr><td>0</td><td>EMD1</td><td>0</td><td>R/W</td></tr>
</table>

MCU Control Register (MCUCR)

The MCU Control register controls the effect of the SLEEP instruction (see AVR instruction set). Note that when the AVR is in sleep mode (stand-by or power-down mode), it will wake up on any enabled interrupt or on any USB activity (if the USB core is activated). Also, on any JTAG activity, if the system is in the power-down mode, it switches into stand-by mode.

addr $35 ($55) 8 bits

Bit	Field	Default	AVR	Description
7	—	0	—	Reserved
6	SE:Sleep Enable	0	R/W	When set, this permits the MCU to enter in the sleep mode when the SLEEP instruction is executed
5	SM:Sleep Mode select bit	0	R/W	This bit selects between the two available sleep modes When the MCU enters into the sleep state and the SM bit is cleared, then it enters into Idle Mode and only the AVR clock is stopped. Otherwise, if the SM bit is set, then the MCU enters into the power- down mode and the MCU disables the oscillator, stopping all clocks and any activity
4:0	—	0	—	Reserved

MCU Status Register (MCUSR)

The MCU Status register provides information on which reset source caused an MCU reset.

addr $34 ($54) 8 bits

Bit	Field	Default	AVR	Description
7:2	—	—	—	—
1	EXTRF: External Reset Flag	See Table 7-2	R/W	External Reset Flag. This flag indicates that an external reset has occurred
0	PORF: Power on Reset Flag		R/W	This flag indicates that a power-on reset has occurred

Table 7-2. PORF and EXTRF Values After Reset

Reset Source	PORF	EXTRF
Power-on Reset	1	Undefined
External Reset	Unchanged	1
Watchdog Reset	Unchanged	Unchanged

The user program must clear these bits as early as possible. If these bits are cleared before a reset condition occurs, the source of resource can be found by using the truth table shown in Table 7-3.

Table 7-3. Reset Source Identification

PORF	EXTRF	Reset Source
0	0	Watchdog Reset
0	1	External Reset
1	x	Power on Reset

External Interrupt Mask Register (EIMSK)

The External Interrupt Mask register masks the external interrupts. External Interrupts should be acknowledged using general-purpose output pins.

addr $37 ($57) 8 bits

Bit	Field	Default	AVR	Description
7	POL3: Polarity of external interrupt 1	0	R/W	INT3 is active high when this bit is low
6	POL2: Polarity of external interrupt 1	0	R/W	INT2 is active high when this bit is low
5	POL1: Polarity of external interrupt 1	0	R/W	INT1 is active high when this bit is low

Bit	Field	Default	AVR	Description
4	POL0: Polarity of external interrupt 0	0	R/W	INT0 is active high when this bit is low
3	INT3	0	R/W	If this is set and the I-bit in the Status register is set, the external pin interrupt 3 is enabled
2	INT2	0	R/W	If this is set and the I-bit in the Status register is set, the external pin interrupt 2 is enabled
1	INT1	0	R/W	If this is set and the I-bit in the Status register is set, the external pin interrupt 1 is enabled
0	INT0	0	R/W	If this is set and the I-bit in the Status register is set, the external pin interrupt 0 is enabled

Clock Control Register (CLK_CNTR)

The Clock Control register controls the clock generation circuit. For example, the PLL output clock rate can be switched from 96 MHz to 192 MHz by setting the CLK_CNTR register to $4C.

addr $0C ($2C) 8 bits

Bit	Field	Default	AVR	Description
7	NPICP	0	R/W	Is equal to the reversed PLL ICP bit. Controls the PLL Charge-Pump current
6	PLCK	0	R/W	If this is set, the PLL Lock signal is used for ‘PLL Stable’ indication. If cleared, the ‘PLL Stable’ indication is equal to the ‘PLL Enable’ signal delayed by a significant factor
5	UDPLL	0	R/W	Select DPLL96 instead of DPLL48 for USB clock recovery
4	OSC_NSLP	0	R/W	If this is set, then do not close the oscillator during the power-down sleep mode
3	PIVCO1	0	R/W	PLL IVCO[1]. Selects the PLL frequency range. Normally, this bit must be equal to the MUL16 bit
2	MUL16	0	R/W	Selects the multiplier of the PLL. When this is set, the external 12 MHz crystal frequency is multiplied by 16 to generate an internal fast clock of 192 MHz. When cleared, the external 12 MHz crystal frequency is multiplied by 8, generating an internal fast clock of 96 MHz
1	MCSP1	0	R/W	AVR core speed select bits. These bits control the AVR clock divisor according to Table 7-4
0	MCSP0	0	R/W	

Table 7-4. Microcontroller Speed Select Bits

MUL16	MCSP1	MCSP0	AVR clock rate	Notes
0	0	0	24 MHz	96 div 4
0	0	1	19, 2 MHz	96 div 5
0	1	0	16 MHz	96 div 6
0	1	1	Reserved	
1	0	0	48 MHz	192 div 4

Table 7-4. Microcontroller Speed Select Bits

MUL16	MCSP1	MCSP0	AVR clock rate	Notes
1	0	1	38,4 MHz	192 div 5
1	1	0	32 MHz	192 div 6
1	1	1	Reserved	

Peripheral Enable Control Register (PERIPHEN)

The Peripheral Enable Control register enables the peripheral components of the system. It controls the SPI, UARTs, USB, and external memory controllers.

addr $0B ($2B) 8 bits

Bit	Field	Default	AVR	Description
7:6	—	0	R	This bit is always read as 0
5	SPI: SPI Enable	0	R/W	When set, this bit enables the SPI controller
4	UART1 Enable	0	R/W	When set, this bit enables the function of UART1
3	UART0 Enable	0	R/W	When set, this bit enables the function of UART0
2	UATTACH	0	R/W	When set, this bit activates (logic 1) the USB_ATTACH pin
1	USB Enable	0	R/W	When set, this bit enables the function of USB (clock enable)
0	EMIEN: External Memory Interface Enable	0	R/W	When set, this bit enables the external memory interface and the pin functions of Port A and Port C[(1)] are set to their alternative pin functions. The EMEN bit overrides any bit direction settings in the respective data direction registers

Program Mode Register (PMOD)

The Program Mode register returns the value of the PMODE0 and PMODE1 input pins.

addr $1F ($3F) 8 bits

Bit	Field	Default	AVR	Description
7:2	—	0	R	Always read as 0
1	PMODE1	N/A	R	The value of the PMODE1 input pin
0	PMODE0	N/A	R	The value of the PMODE0 input pin

I/O Debug Register (IDR)

The I/O Debugs register communicates between the on-chip debug system (through JTAG) and the AVR CPU. It provides a communication channel from the running program in the microcontroller to the debugger. The CPU can transfer a byte to the debugger by writing to this location.

addr $3C ($5C) 8 bits

Bit	Field	Default	AVR	Description
7	IDRD:I/O Debug Register Dirty	0	R/W	This bit is set to indicate to the debugger that the register has been written
6		0	R/W	When the CPU reads the IDR register, the 7 LSB will be from the IDR register, while the MSB is from the IDRD bit. The debugger clears the IDRD bit after it has read the information
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

IrDA Control Register (IRDACR)

The IrDA Control register controls the IrDA 1.0 codec. The IrDA codec is connected between the UART0 and UART1 SIN and SOUT signals and the corresponding pins.

addr $22 ($42) 8 bits

Bit	Field	Default	AVR	Description
7:5	—	0	R	
4	TXPOL: Transmit Polarity	0	R/W	If this bit is set, the SOUT signal is inverted
3	RXPOL: Receive Polarity	0	R/W	If this bit is set, the SIN signal is inverted before entering the IrDA codec
2	MODE	0	R/W	When this bit is cleared, it enables the 3/16 return-to-zero encoding scheme. When this bit is set, it enables the 4/16 return-to-zero encoding scheme
1	USEL: UART Select	0	R/W	When this bit is cleared, the codec uses UART0. When this bit is set, the codec uses UART1
0	IRDAEN	0	R/W	When this bit is set, it enables the IrDA codec. When this bit is cleared, the codec is transparent to the UART0/1 SIN/SOUT pins

7.1 Timers/Counters

Timer/Counter Interrupt Mask Register (TIMSK)

The Timer/Counter Interrupt Mask register masks the internal timer interrupts.

addr $39 ($59) 8 bits

Bit	Field	Default	AVR	Description
7	TOIE1: Timer/Counter 1 Overflow Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $001C) is executed if an overflow in Timer/Counter1 occurs. The Timer/Counter1 Overflow Flag is set in the Timer/Counter1 Interrupt Flag register (TIFR)
6	OCIE1A: Timer/Counter 1 Output Compare A Match Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Timer/Counter1 Compare A Match interrupt is enabled. The corresponding interrupt (at vector $0018) is executed if a Compare A match in Timer/Counter1 occurs. The Compare A flag in Timer/Counter 1 is set, in the Timer/Counter Interrupt flag register (TIFR)
5	OCIE1B: Timer/Counter 1 Output Compare B Match Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Timer/Counter1 Compare B Match interrupt is enabled. The corresponding interrupt (at vector $001A) is executed if a Compare B match in Timer/Counter1 occurs.The Compare B flag in Timer/Counter 1 is set, in the Timer/Counter Interrupt flag register (TIFR)
4	—	0	R	
3	TICIE1: Timer/Counter 1 Input Capture Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Input Capture Event interrupt is enabled. The corresponding interrupt (at vector $0016) is executed if a capture event occurs on pin PB3. The Input Capture Flag in Timer/Counter1 is set in the Timer/Counter Interrupt Flag Register (TIFR)
2	TOIE2: Timer/Counter 2 Overflow Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Timer/Counter2 Overflow interrupt is enabled. The corresponding interrupt (at vector $0020) is executed if an overflow in Timer/Counter2 occurs. The Timer/Counter2 Overflow Flag is set in the Timer/Counter2 Interrupt Flag Register (TIFR)
1	TOIE0: Timer/Counter 0 Overflow Interrupt Enable	0	R/W	When this bit is set and the I-bit in the Status register is 1, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $0020) is executed if an overflow in Timer/Counter0 occurs. The Timer/Counter0 Overflow Flag is set in the Timer/Counter0 Interrupt Flag Register (TIFR)
0	—	0	R	

Timer/Counter Interrupt Flag Register (TIFR)

addr $38 ($58) 8 bits

Bit	Field	Default	AVR	Description
7		0	R/W	
6	OCFA: Output Compare Flag A	0	R/W	The OCFA is set when a compare match between Timer/Counter1 and the OCR1A register occurs. This flag is cleared when written with a logic 1
5	OCFB: Output Compare Flag 1B	0	R/W	The OCF1B is set when a compare match between Timer/Counter1 and the OCR1B register occurs. This flag is cleared when written with a logic 1
4	—	0	R	
3	ICF1: Input Capture Flag	0	R/W	This flag when set indicates an input capture event, where the contents of the Timer/Counter1 are transferred to the ICR1 register. This flag is cleared when written with a logic 1
2	TOV2: Timer/Counter 2 Overflow Flag	0	R/W	Timer/Counter2 Overflow Flag. The TOV2 is set when an overflow occurs in Timer/Counter2. This flag is cleared when written with a logic 1
1	TOV0: Timer/Counter 1 Overflow Flag.	0	R/W	Timer/Counter0 Overflow Flag. The TOV0 is set when an overflow occurs in Timer/Counter0. This flag is cleared when written with a logic 1
0	—	0	R	

7.2 Timer/Counter 0 and Timer/Counter 2

Timer/Counter 0 Control Register (TCCR0)

addr $33 ($53) 8 bits

Bit	Field	Default	AVR	Description
7:3	—	0	R	

Bit	Field	Default	AVR	Description
2	CS02	0	R/W	
1	CS01		R/W	
0	CS00		R	

CS02	CS01	CS00	Description
0	0	0	Stop, Timer/Counter0 is stopped
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0 (orT2), Falling Edge
1	1	1	External Pin T0 (or T2), Rising Edge

The Timer/Counter2 Control Register (TCCR2)

addr $25 ($45) 8 bits

Bit	Field	Default	AVR	Description
7:3	Reserved	0	R	—
2	CS02: Clock Select 0 bit 2	0	R/W	The Clock Select0 bits 2,1, and 0 define the pre-scaling source of Timer0
1	CS01: Clock Select 0 bit 1	0	R/W	
0	CS00: Clock Select 0 bit 0	0	R/W	

CS02	CS01	CS00	Description
0	0	0	Stop, Timer/Counter0 is stopped
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0 (or T2), Falling Edge
1	1	1	External Pin T0 (or T2), Rising Edge

Timer/Counter0 Register (TCNT0)

addr $32 ($52) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	Both Timer/Counter0 and Timer/Counter2 are realized as an up-counter with read and write access. If the TCNT0 (or TCNT2 respectively) is written and a clock source is present, the Timer/Counter0 (or Timer/Counter2) continues counting in the clock cycle following the write operation
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

Timer/Counter2 Register (TCNT2)

Both Timer/Counter0 and Timer/Counter2 are realized as an up-counter with read and write access. If the TCNT0 (or TCNT2 respectively) is written and a clock source is present, the Timer/Counter0 (or Timer/Counter2) continues counting in the clock cycle following the write operation.

addr $24 ($44) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	Both Timer/Counter0 and Timer/Counter2 are realized as an up-counter with read and write access. If the TCNT0 (or TCNT2 respectively) is written and a clock source is present, the Timer/Counter0 (or Timer/Counter2) continues counting in the clock cycle following the write operation
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

Pre-load Register 0 (PRELD0)

addr $31 ($51) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	An 8-bit R/W register with a 0 initial value. The contents of this register are loaded to Timer/Counter0 TCNT0 (or Timer/Counter2 TCNT2 respectively) after an overflow
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

Pre-load Register 2 (PRELD2)

addr $23 ($43) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	An 8-bit R/W register with 0 initial value. The contents of this register are loaded to Timer/Counter0 TCNT0 (or Timer/Counter2 TCNT2 respectively) after an overflow
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

7.3 Timer/Counter 1

Timer/Counter1 Register (TCCR1B)

addr $2E ($4E) 8 bits

Bit	Field	Default	AVR	Description
7	ICNC1: Input Capture 1 Noise Canceler (4 CKs	0	R/W	When this bit is 0, the input canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the Input Capture Pin (ICP), as specified by the ICES1 bit. When this bit is set, four successive samples are measured and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is the CPU clock frequency
6	ICES1: Input Capture1 Edge Select	0	R/W	When this bit is cleared, the Timer/Counter1 contents are transferred to the Input Capture register (ICR1) on the falling edge of the ICP. When this bit is 1, the contents are transferred on the rising edge
5:4	—	0	R	
3	CTCA1: Clear Timer/Counter 1 on Compare A match	0	R/W	When this bit is 1, the Timer/Counter1 is reset to $0000 after compare A match. If this bit is cleared, the Timer/Counter1 continues counting after a compare A match
2	CS12: Clock Select 1,bit 2	0	R/W	These bits select prescaling source for the Timer/Counter,1 according to the following table
1	CS11:Clock Select 1, bit 1	0	R/W	
0	CS11:Clock Select 0, bit 0	0	R/W	

CS02	CS01	CS00	Description
0	0	0	Stop, Timer/Counter0 is stopped
0	0	1	CK
0	1	0	CK/8
0	1	1	CK/64
1	0	0	CK/256
1	0	1	CK/1024
1	1	0	External Pin T0 (orT2), Falling Edge
1	1	1	External Pin T0 (or T2), Rising Edge

Timer/Counter1 (TCNT1H and TCNT1L)

The Timer/Counter1 is realized as a 16-bit up counter consisted of two 8-bit registers TCNT1H and TCNT1L. These registers have read and write access with an initial value of $00. If the Timer/Counter1 (TCNT1H and TCNT1L) register is written to and a clock source is selected, the Timer/Counter1 continues counting in the timer clock cycle after it is preset with the written value.

To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP). This temporary register is also used when accessing OCR1A, OCR1B, and ICR1. If the main program and also interrupt routines perform access to registers using TEMP, the interrupts must be disabled during access from the main program.

addr $2D ($4D), $2c ($4C) 16 bits

Bit	Field	Default	AVR	Description
15	MSB	0	R/W	**TCNT1 Timer/Counter1 Write**: When the CPU writes to the high byte TCNT1H, the written data is placed in the TEMP register. Next, when the CPU writes the low byte TCNT1L, this byte of data is combined with the TEMP register and all 16-bits are written simultaneously to the Timer/Counter1 TCNT1 register. Consequently, the high byte must be accessed first for a full 16-bit write operation. When using the Timer/Counter1 as an 8-bit counter, it is sufficient to write the low byte only **TCNT1 Timer/Counter1 Read**: When the CPU reads the low byte TCNT1L, the data are placed in the TEMP register. Next, when the CPU reads the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte must be accessed first for a full 16-bit read operation. When using Timer/Counter1 as an 8-bit counter, it is sufficient to read the low byte only
14		0	R/W	
13		0	R/W	
12		0	R/W	
11		0	R/W	
10		0	R/W	
9		0	R/W	
8		0	R/W	
7		0	R/W	
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

Timer/Counter1 Output Compare Register A (OCR1AH and OCR1AL)

Timer/Counter Output Compare register A consists of 16 bits and is made by two 8-bit R/W registers, with the initial value of 0, namely the OCR1AH and OCR1AL.

Full 16-bit write and read operations are made according to the way specified for the Timer/Counter1 TCNT1.

addr $2B ($4B), $2A ($4A) 16 bits

Bit	Field	Default	AVR	Description
15	MSB	0	R/W	OCR1AH
14:8		0	R/W	
7:1		0	R/W	OCR1AL
0	LSB	0	R/W	

Timer/Counter1 Output Compare Register B (OCR1BH and OCR1BL)

Timer/Counter Output Compare register B consists of 16 bits and is made by two 8-bit R/W registers, with the initial value of 0, namely the OCR1BH and OCR1BL.

Full 16-bit write and read operations are made according to the way specified for the Timer/Counter1 TCNT1.

addr $29 ($49), $28 ($48) 16 bits

Bit	Field	Default	AVR	Description
15	MSB	0	R/W	OCR1BH
14:8		0	R/W	
7:1		0	R/W	OCR1BL
0	LSB	0	R/W	

The Timer/Counter1 Input Capture Register (ICR1H and ICR1L)

The Timer/Counter Input Compare register consists of 16-bits and is made by two 8-bit read only registers, with initial value of 0, namely the ICR1H and ICR1L.

When the rising or falling edge (according to the Input Capture Edge Setting (ICES1) of the signal at the Input Capture Pin (ICP) is detected, the current value of the Timer/Counter1 is transferred to the Input Capture register (ICR1). At the same time, the Input Capture Flag (ICF1) is set to 1.

Full 16-bit read operations are made according to the way specified for the Timer/Counter1 TCNT1 above.

addr $27 ($47), $26 ($46) 16 bits

Bit	Field	Default	AVR	Description
15	MSB	0	R	OCR1H
14:8		0	R	
7:1		0	R	OCR1L
0	LSB	0	R	

7.4 Watchdog Timer

Watchdog Timer Control Register (WDTCR)

addr $21 ($41) 8 bits

Bit	Field	Default	AVR	Description
7:5	—	0	R	
4	WDTOE: Watchdog Turn Off Enable	0	R/W	This bit must be set when the WDE bit is cleared. Otherwise, the Watchdog Timer will not be disabled. Once set, H/W will clear this bit to 0 after four clock cycles
3	WDE: Watchdog Enable	0	R/W	When this bit is set, the Watchdog Timer is enabled. When this bit is 0, the Watchdog Timer is disabled. The WDE bit can only be cleared if the WDTOE bit is set. To disable an enabled Watchdog Timer, the following procedure must be followed: In the same operation, write a logical 1 to bits WDTOE and WDE. A logical 1 must be written to the WDE bit even though it is set to 1 before the disable operation starts. Within the next four clock cycles, write a logical 0 to bit WDE. This disables the watchdog.
2	WDP2: Watchdog Timer Prescaler 2	0	R/W	These bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled, according to Table 7-5
1	WDP1:Watchdog Timer Prescaler 1		R/W	
0	WDP0:Watchdog Timer 0 Prescaler 0		R/W	

Table 7-5. Watchdog Timer Prescale Register Select

WDP2	WDP1	WDP0	Time-out Period	
0	0	0	5.5 ms	16K cycles
0	0	1	11 ms	32K cycles
0	1	0	22 ms	64K cycles
0	1	1	47 ms	128K cycles
1	0	0	87 ms	256K cycles
1	0	1	175 ms	512K cycles
1	1	0	350 ms	1024K cycles
1	1	1	700 ms	2084K cycles

7.5 SPI Interface

SPI Control Register (SPCR)

addr $0D ($2F) 8 bits

Bit	Field	Default	AVR	**Description**
7	SPIE: SPI Interrupt Enable	0	R/W	This bit causes the setting of the SPIF bit in the SPSR register to execute the SPI interrupt provided that the global interrupt is enabled
6	SPE: SPI Enable	0	R/W	When this bit is set, the SPI is enabled and SS, MOSI, MISO, and SCK are connected to pins PB4, PB5, PB6, and PB7
5	DORD: Data Order	0	R/W	When this bit is 1, the LSB of the data word is transmitted first. When this bit is cleared, the MSB of the data word is transmitted first
4	MSTR: Master/Slave Select	0	R/W	This bit selects the master SPI when set and the slave SPI mode when cleared. If the SS is configured as input and driven low while the MSTR is set, the MSTR will be cleared, and the SPIF in the SPSR will become set. The user will then have to set the MSTR to re-enable the SPI master mode
3	CPOL: Clock Polarity	0	R/W	When this bit is set, SCK is high when idle. When the CPOL is cleared, the SCK is low when idle
2	CPHA: Clock Phase	0	R/W	When this bit is set, the data is valid in the falling edge of the SCK if the CPOL = 0, or in the rising edge of the SCK when the CPOL = 1. When this bis is cleared, the data is valid in the rising edge of the SCK if the CPOL = 0 and in the falling edge of the SCK if the CPOL = 1
1	SPR1: SPI Clock Rate Select 1	0	R/W	These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect during the slave mode. The relationship between the slave and the AVR clock frequency is shown in the following table:
0	SPR0: SPI Clock Rate Select 0	0	R/W	

SPR1	SPR0	SCG Frequency
0	0	{AVR clock rate} div 4
0	1	{AVR clock rate} div 16
1	0	{AVR clock rate} div 64
1	1	{AVR clock rate} div 128

SPI Status Register (SPSR)

addr $0E ($2E) 8 bits

Bit	Field	Default	AVR	Description
7	SPIF: SPI Interrupt Flag	0	R/W	When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE in SPCR is set and the global interrupts are enabled. Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set, then accessing the SPI Data register.
6	WCOL: Write Collision Flag	0	R/W	The WCOL bit is set if the SPI Data register (SPDR) is written during a data transfer. The WCOL and the SPIF bits are cleared by first reading the SPI Status register with the WCOL set, and then by accessing the SPI Data register
5:0	—	0	R	Always read as 0

SPI Data Register (SPDR0

addr $0F ($2F) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	It is used for data transfer between the register file and the SPI Shift register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read
6:1		0	R/W	
0	LSB	0	R/W	

7.6 I/O PORTs

7.6.1 PORT A

Port A is an 8-bit bi-directional I/O Port with internal pull-up resistors.

7.6.1.1 PORT A Alternative Functionality

Port A, besides its use as a general-purpose I/O port, is used to support alternate functions. Specifically, serves also as the Address and Data bus of the External Memory Interface AD[0-7].

7.6.1.2 PORT A I/O Registers

PORT A Data Register (PORTA)

addr $1B ($3B) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	It is an 8-bit R/W register with zero initial value 0xff. The bits in the Data Direction Register control the direction of the corresponding pin in the PORT A. When bit DDRBx is set, then PBx pin is input, while when DDRBx is cleared PBx is output, x=0…7
6:1		0	R/W	
0	LSB	0	R/W	

PORT A Data Direction Register (DDRA)

addr $1A($3A) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The bits in the Data Direction register control the direction of the corresponding pin in the PORT A. When bit DDRAx is set ,then the PAx pin is output, while when DDRAx is cleared, the PAx pin is input (X = 0..7)
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

PORT A Input Pins Address (PINA)

addr $19($39) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	N/A	R	The Port A Input Pins Address (PINA) is not a physical register. This address enables access to the physical voltage value on each port pin. It is a read-only address. The initial value of PINA depends from status of each PORT A pin
6		N/A	R	
5		N/A	R	
4		N/A	R	
3		N/A	R	
2		N/A	R	
1		N/A	R	
0	LSB	N/A	R	

7.6.2 PORT B

PORT B is an 8-bit bi-directional I/O port with internal pull-up resistors.

7.6.2.1 PORT B Alternative Functionality

Port B, besides its use as a general-purpose I/O port, is used to support alternate functions. Specifically, the SPI interface, the input capture pin for Timer/Counter1, and the external clocks for the Timers/Counters are implemented through the B Port. When the SPI is enabled, the data direction of port B pins 4..7 is overridden, according to Table 7-6.

Table 7-6. Port B Pins Alternate Functions

Port Pin	Direction		**Alternate Function**
PB0	(DDRB)		External clock pin for Timer/Counter0
PB1	(DDRB)		External clock pin for Timer/Counter1
PB2	(DDRB)		Input capture pin for Timer/Counter1
PB3	(DDRB)		ICP: Capture for Timer/Counter 1
	SPI Master	SPI Slave	

Table 7-6. Port B Pins Alternate Functions

Port Pin	Direction		**Alternate Function**
PB4	(DDRB)	Input	nSS (SPI slave select input)
PB5	Output	Input	MOSI (SPI bus master output/slave input)
PB6	Input	Output/Hi-Z	MISO (SPI bus master input/slave output)
PB7	Output	Input	SCK (SPI bus serial clock)

7.6.2.2 PORT B I/O Registers

PORT B Data Register (PORTB)

addr $18 ($38) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	It is an 8-bit R/W register with a 0 initial value 0xff. The bits in the Data Direction register control the direction of the corresponding pin in PORT B When bit DDRBx is set, then PBx pin is input, while when DDRBx is cleared, the PBx is output (x = 0…7)
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

PORT B Data Direction Register (DDRB)

addr $17 ($37) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The bits in the Data Direction register control the direction of the corresponding pin in PORT B. When bit DDRBx is set, then the PBx pin is output, while when DDRBx is cleared, the PBx pin is input (X = 0..7)
6		0	R/W	
5		0	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

PORT B Input Pins Address (PINB)

addr $16 ($36) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	N/A	R	The Port B Input Pins address (PINB) is not a physical register. This address enables access to the physical voltage value on each port pin. It is a read-only address. The initial value of PINB depends from status of each PORT B pin
6		N/A	R	
5		N/A	R	
4		N/A	R	
3		N/A	R	
2		N/A	R	
1		N/A	R	
0	LSB	N/A	R	

7.6.3 PORT C

PORT C is a 8-bit Output Port with internal pull-up resistors.

7.6.3.1 PORT C Alternative Functionality

In addition to functioning as a general-purpose I/O port, PORT C is used to support alternate functions. Specifically, it is used as the Address bus A[8-14]and ALE of the external memory interface.

7.6.3.2 PORT C I/O Regsiters

PORT C Data Register (PORTC)

addr $15 ($35) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	It is an 8-bit R/W register with zero initial value 0xff. The bits in the Data Direction register control the direction of the corresponding pin in the PORT C When bit DDRBx is set, then PBx pin is input, while when DDRBx is cleared PBx is output, x = 0…7
6:1		0	R/W	
0	LSB	0	R/W	

PORT C Data Direction Register

addr $14 ($34) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The bits in the Data Direction register controls the direction of the corresponding pin in the PORT C When bit DDRCx is set, then PCx pin is output, while when the DDRCx is cleared, the PCx pin is input (x = 0..7)
6:1		0	R/W	
0	LSB	0	R/W	

PORT C Input Pins Address (PINC)

addr $13 ($33) 8 bits

Bit	Field	Default	AVR	**Description**
7	MSB	N/A	R	The Port C Input Pins address (PINC) is not a physical register. This address enables access to the physical voltage value on each port pin. It is a read-only address. The initial value of PINC depends on the status of each PORT C pin
6:1		N/A	R	
0	LSB	N/A	R	

7.6.4 PORT D

PORT D is an 8-bit bi-directional I/O port with internal pull-up resistors.

7.6.4.1 PORT D Alternative Functionality

In addition to functioning as a general-purpose I/O port, PORT D is used to support alternate functions. Specifically, it offers the signals of UART0, as shown in Table 7-7. Also, when the UART0 is enabled, the data direction of the port D pins is overridden, according to Table 7-7.

PORT D pins use the special pad “Vdd for PORTD” for a power supply.

Table 7-7. Port D Pins Alternate Functions

Port Pin	Direction	**Alternate Function**
PD0	Input	Serial Receive In UART0
PD1	Output	Serial Transmit Out UART0
PD2	Bi-directional	nRTS, UART0 Ready To Send
PD3	Bi-directional	nCTS, UART0 Clear To Send
PD4	Bi-directional	nDSR, UART0 Data Set Ready
PD5	Bi-directional	nDTR, UART0 Data Terminal Ready
PD6	Bi-directional	nCD, UART0 Carrier Detect
PD7	Bi-directional	nRI, UART0 Ring Indicator

7.6.4.2 PORT D I/O Registers

PORT D Data Register (PORTD)

addr $12 ($32) 8 bits

Bit	**Field**	**Default**	**AVR**	**Description**
7	MSB	0	R/W	It is an 8-bit R/W register with zero initial value 0xff. The bits in the Data Direction register control the direction of the corresponding pin in the PORT D. When bit DDRBx is set, then PBx pin is input, while when DDRBx is cleared PBx is output (x = 0…7)
6:1		0	R/W	
0	LSB	0	R/W	

PORT D Data Direction Register

addr $11 ($31) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The Data Direction register controls the direction of the corresponding pin in the PORT D When bit DDRDx is set, then the PDx pin is input, while when DDRDx is cleared, the PDx pin is output (x = 0…7)
6:1		0	R/W	
0	LSB	0	R/W	

PORT D Input Pins Address (PIND)

addr $10 ($30) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	N/A	R	The Port D Input Pins address (PIND) is not a physical register. This address enables access to the physical voltage value on each port pin. It is a read-only address. The initial value of PIND depends on the status of each PORT D pin
6:1		N/A	R	
0	LSB	N/A	R	

7.6.5 PORT E

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors at pins PE0, PE1, and PE4..7, and internal pull-down resistors at pins PE2 and PE3.

7.6.5.1 PORT E Alternative Functionality

In addition to functioning as a general-purpose I/O port, PORT E is also used to support alternate functions. Specifically, serves also as the Tx/Rx signals of UART1, as external interrupts and read/write signals for the External Memory Interface. When an alternative function is enabled, the data direction of port E pins is overridden, according to Table 7-8.

Table 7-8. Port E Pins Alternate Functions

Port Pin	Direction	Alternate Function
PE0	Input	Serial Receive In UART1
PE1	Output	Serial Transmit Out UART1
PE2	(DDRE)	INT0: Edge triggered or level sensitive interrupt with pull-down
PE3	(DDRE)	INT1: Edge triggered or level sensitive interrupt with pull-down
PE4	(DDRE)	INT2: Edge triggered or level sensitive interrupt with pull-up
PE5	(DDRE)	INT3: Edge triggered or level sensitive interrupt with pull-up
PE6	Output	nWR: Write signal for the external memory interface
PE7	Output	nRD: Read signal for the external memory interface

7.6.5.2 PORT E I/O Registers

PORT E Data Register (PORTE)

addr $0A ($2A) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The bits in the Data Direction register control the direction of the corresponding pin in the PORT E When bit DDRBx is set, then PBx pin is input, while when DDRBx is cleared PBx is output, x = 0…7
6:1		0	R/W	
0	LSB	0	R/W	

PORT E Data Direction Register (DDRE)

addr $09 ($29) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	The bits in the Data Direction register control the direction of the corresponding pin in the PORT E When bit DDREx is set, then PEx pin is output, When DDREx is cleared, the PEx pin is input (x = 0..7).
6:1		0	R/W	
0	LSB	0	R/W	

PORT E Input Pins Address (PINE)

addr $08 ($28) 8 bits

Bit	Field	Default	AVR	Description
7	MSB	N/A	R	The Port E Input Pins address (PINE) is not a physical register. This address enables access to the physical voltage value on each port pin. It is a read-only address. The initial value of PINE depends from status of eachPORT E pin
6:1		N/A	R	
0	LSB	N/A	R	

8. Memory Space (Register Description)

8.1 USB Register Set

The USB appears to the AVR to be just another peripheral. The USB register file is mapped to the SRAM space. Table 8-1 summarizes the USB cell specific registers.

Table 8-1. USB Register Set

Register	Address	Default	Function
SLP_MODE	$F000	00000000b	Sleep mode control
IRQ_EN	$F001	00000000b	Master interrupt enable
IRQ_STAT	$F002	00000000b	Master interrupt status
RES_STAT	$F003	00000000b	Reset status
PAIR_EN	$F004	00000000b	Pair addressing enable
USB_DMA_ADL	$F005	00000000b	DMA address Low

Table 8-1. USB Register Set (Continued)

Register	Address	Default	Function
USB_DMA_ADH	$F006	00000000b	DMA address High
USB_DMA_LEN	$F007	00000000b	DMA packet length requested
USB_DMA_EAD	$F008	00000000b	DMA target Endpoint address
USB_DMA_PLT	$F009	00000000b	DMA packet length transferred
USB_DMA_EN	$F00A	00000000b	DMA enable
FBYTE_CNT7_H	$F0A8	00000000b	FIFO Byte Count 7 register [10:8]
FBYTE_CNT6_H	$F0A9	00000000b	FIFO Byte Count 6 register [10:8]
FBYTE_CNT5_H	$F0AA	00000000b	FIFO Byte Count 5 register [10:8]
FBYTE_CNT4_H	$F0AB	00000000b	FIFO Byte Count 4 register [10:8]
FBYTE_CNT3_H	$F0AC	00000000b	FIFO Byte Count 3 register [10:8]
FBYTE_CNT2_H	$F0AD	00000000b	FIFO Byte Count 2 register [10:8]
FBYTE_CNT1_H	$F0AE	00000000b	FIFO Byte Count 1 register [10:8]
FBYTE_CNT0_H	$F0AF	00000000b	FIFO Byte Count 0 register [10:8]
FBYTE_CNT7_L	$F0B8	00000000b	FIFO Byte Count 7 register [7:0]
FBYTE_CNT6_L	$F0B9	00000000b	FIFO Byte Count 6 register [7:0]
FBYTE_CNT5_L	$F0BA	00000000b	FIFO Byte Count 5 register [7:0]
FBYTE_CNT4_L	$F0BB	00000000b	FIFO Byte Count 4 register [7:0]
FBYTE_CNT3_L	$F0BC	00000000b	FIFO Byte Count 3 register [7:0]
FBYTE_CNT2_L	$F0BD	00000000b	FIFO Byte Count 2 register [7:0]
FBYTE_CNT1_L	$F0BE	00000000b	FIFO Byte Count 1 register [7:0]
FBYTE_CNT0_L	$F0BF	00000000b	FIFO Byte Count 0 register [7:0]
FDR7	$F0C8	00000000b	FIFO 7 Data register
FDR6	$F0C9	00000000b	FIFO 6 Data register
FDR5	$F0CA	00000000b	FIFO 5 Data register
FDR4	$F0CB	00000000b	FIFO 4 Data register
FDR3	$F0CC	00000000b	FIFO 3 Data register
FDR2	$F0CD	00000000b	FIFO 2 Data register
FDR1	$F0CE	00000000b	FIFO 1 Data register
FDR0	$F0CF	00000000b	FIFO 0 Data register
ECSR7	$F0D8	x1110000b	Endpoint7 Control and Status register
ECSR6	$F0D9	x1110000b	Endpoint6 Control and Status register
ECSR5	$F0DA	x1110000b	Endpoint5 Control and Status register
ECSR4	$F0DB	x1110000b	Endpoint4 Control and Status register
ECSR3	$F0DC	x1110000b	Endpoint3 Control and Status register
ECSR2	$F0DD	x1110000b	Endpoint2 Control and Status register
ECSR1	$F0DE	x1110000b	Endpoint1 Control and Status register

Table 8-1. USB Register Set (Continued)

Register	Address	Default	Function
ECSR0	$F0DF	x1110000b	Endpoint0 Control and Status register
ECR7	$F0E8	0xxx0000b	Endpoint7 Control register
ECR6	$F0E9	0xxx0000b	Endpoint6 Control register
ECR5	$F0EA	0xxx0000b	Endpoint5 Control register
ECR4	$F0EB	0xxx0000b	Endpoint4 Control register
ECR3	$F0EC	0xxx0000b	Endpoint3 Control register
ECR2	$F0EC	0xxx0000b	Endpoint2 Control register
ECR1	$F0EE	0xxx0000b	Endpoint1 Control register
ECR0	$F0EF	0xxx0000b	Endpoint0 Control register
ENDPPGPG	$F0F1	00000000b	Function Endpoint Ping-pong register
FADDR	$F0F2	00000000b	Function Address register
UIER	$F0F3	xxx00000b	USB Interrupt Enable register
UIAR	$F0F5	xxxxx000b	USB Interrupt Acknowledge register
UISR	$F0F7	00000000b	USB Interrupt Status register
SPRSIE	$F0F9	xxxxx000b	Suspend/Resume Interrupt Enable register
SPRSR	$F0FA	xxxxx000b	Suspend/Resume register
GLB_STATE	$F0FB	xxxxx000b	Global State register
FRM_NUM_L	$F0FC	xxxxx000b	Frame Number Low register
FRM_NUM_H	$F0FD	xxxxx000b	Frame Number High register

SLP_MODE (Sleep Mode Control Register)

addr: $F000 8 bits

Bit	Field	AVR	Description
7:6	Reserved		Reserved and set to 0
5	SLP	R/W	If set, put the USB Controller in sleep mode
4:0	Reserved		Reserved and set to 0

IRQ_EN (USB Interrupt Mask Register)

addr: $F001 8 bits

Bit	Field	AVR	Description
7	Reserved	R/W	Reserved and set to 0
6	INT_EN	R/W	When this bit is high, enables the USB protocol handler to cause an interrupt (see UISR and IRQ_STAT[6])

Bit	Field	AVR	Description
5:2	Reserved	R/W	Reserved and set to 0
1	SUSP_INT_EN	R/W	If this bit is high, an interrupt is generated when the USB enters suspend mode A USB device enters in suspend mode only when requested by the USB Host through bus inactivity for at least 3 ms
0	RSM_INT_EN	R/W	If this bit is high, an interrupt is generated when the USB enters resume mode. A J-to-K state change on the USB port signal resume

IRQ_STAT (USB Interrupt Status Register)

The IRQ_STAT register is automatically cleared on each read access.

addr: $F002 8 bits

Bit	Field	AVR	Description
7	Reserved	R	Reserved and set to 0
6	INT	R	Interrupt from the USB protocol handler. When this bit is high, then at least one bit of UISR is set
5:4	Reserved	R	Reserved and set to 0
3	USB_RST	R	This bit is high, while the USB bus remains in reset state. This bit can be accessed also by using the RES_STAT register, avoiding to clear the IRQ_STAT
2	nSUSP	R	When this bit is high, the USB has exited from the suspend mode
1	pSUSP	R	When this bit is high, the USB has entered the suspend mode
0	pUSB_RST	R	When this bit is high, the USB host controller has sent a Reset request (USB bus entered the reset state)

RES_STAT (Reset Status)

addr: $F003 8 bits

Bit	Field	AVR	Description
7:4	Reserved	R	Reserved and set to 0
3	USB_RST	R	This bit is high while the USB bus remains in reset state
2:0	Reserved	R	Reserved and set to 0

PAIR_EN (Pair Addressing Enable)

addr: $F004 8 bits

Bit	Field	AVR	Description
7:4	Reserved	R	Reserved and set to 0
3:1	UPA[3:1]	R/W	By setting any of these bits, a pair of EPs is formed of one IN and one OUT. For example, if UPA[1] is set, the EP1 should be configured as an OUT EP, while the EP4 as an IN EP: UPA[1]: EP4 has the same USB physical address with EP1 UPA[2]: EP5 has the same USB physical address with EP2 UPA[3]: EP6 has the same USB physical address with EP3
0	Reserved	R	Reserved and set to 0

USB_DMA_ADL (DMA Address Low)

addr: $F005 8 bits

Bit	Field	AVR	Description
7:0	UDA[7-0]	R/W	The least significant byte of the target address at the Data Memory that the DMA controller will use

USB_DMA_ADH (DMA Address High)

addr: $F006 8 bits

Bit	Field	AVR	Description
7	Reserved	R	Reserved and set to 0
6:0	UDA[14-8]	R/W	These are seven bits along with the eight bits of the USB_DMA_ADL, form the target address UDA[14-0] at the Data Memory that the DMA will use

USB_DMA_LEN (DMA Packet Length)

addr: $F007 8 bits

Bit	Field	AVR	Description
7:0	PLEN[7-0]	R/W	The AVR writes the number of bytes for the next DMA

USB_DMA_EAD (DMA Target Endpoint Address)

addr: $F008 8 bits

Bit	Field	AVR	Description
7:0	EAD[7:0]	R/W	The AVR writes the offset byte of the FDRx address, depending on the Endpoint that is going to send or has received the data: The following Endpoints with the corresponding offset bytes are supported by the DMA channels: FDR1: $CE FDR2: $CD FDR3: $CC FDR4: $CB FDR5: $CA FDR6: $C9 FDR7: $C8

USB_DMA_PLT (DMA Packet Length Transferred)

addr: $F009 8 bits

Bit	Field	AVR	Description
7:0	TPL[7:0]	R/W	Returns the number of bytes transferred during the last DMA

USB_DMA_EN (DMA Enable Register)

addr: $F00A 8 bits

Bit	Field	AVR	Description
7:2	Reserved	R	Reserved and set to 0
1	USB_RDMA_EN	R/W	Enables Receive DMA (for OUT EPs). This bit is automatically cleared after the end of the DMA
0	USB_TDMA_EN	R/W	Enables Transmit DMA (for IN EPs). This bit is automatically cleared after the end of the DMA

FBYTE_CNTx_H (FIFO Byte Count High Register)

Each Endpoint has a register that stores the number of bytes to be sent or that was received by the USB H/W. The maximum data packet supported is 1024 bytes length for isochronous Endpoints.

addr: See Table 8-1

Bit	Field	AVR	Description
7:3	Reserved	Reserved	Reserved
2:0	BYTECNT[10:8]	R/W	Length of data packet in FIFO

FBYTE_CNTx_L (FIFO Byte Count Low Register)

addr: See Table 8-1

Bit	Field	AVR	Description
7:0	BYTECNT[7:0}	R/W	Length of data packet in FIFO

FDR (FIFO Data Registers 0 –7)

FIFO Data registers are dual function buffer registers. Received data are read by the processor from the Endpoint's FIFO through these data registers. In the transmit mode, the processor writes to the FIFO though this register.

addr: See Table 8-1 8 bits

Bit	Field	AVR	Description
7:0	FIFO DATA [7:0]	R/W	Data to be written to FIFO or data to be read from the FIFO

ECSR (Endpoint Control and Status Registers 0 – 7)

addr: See Table 8-1 8 bits

Bit	Field	AVR	Description
7	Control Direction	R	This bit is set by the processor to indicate to the USB H/W the direction of a control transfer 0 = control write. No data stage 1 = control read This bit is used by Control Endpoints only and is used by FW to indicate the direction of a control transfer. It is written by the FW after it receives a RX SETUP interrupt. The H/W uses this bit to determine the status phase of a control transfer
6	Data End	R	This bit indicates that the processor has placed the last data packet in FIFO0, or that the processor has processed the last data packet it expects from the Host This bit is used only by Control Endpoints together with bit 1 (TX Packet Ready) to signal the USB H/W to go to the STATUS phase after the packet currently residing in the FIFO is transmitted After the H/W completes the STATUS phase it will interrupt the processor without clearing this bit CAUTION: Because the Data End bit signals “END OF TRANSACTION”, any other Endpoint controller bit set after the DATA END is not considered by the Ping-pong controller., which is why Tx_Packet Ready should be set before DATA END
5	Force Stall	R	This bit is set by the processor to indicate a stalled Endpoint. The H/W will send a STALL handshake as a response to the next IN or OUT token The processor sets this bit if it wants to force a STALL if an unsupported request is received or if the Host continues to ask for data after the data is exhausted. This bit should be set at the end of any data phase or setup phase

Bit	Field	AVR	Description
4	TX Packet Ready	R/C	This bit indicates that the processor has loaded the FIFO with a packet of data. This bit is cleared by the H/W after the USB Host acknowledges the packet. For ISO Endpoints, this bit is cleared unconditionally after the data is sent This bit is used for the following operations : • Control read transactions by a Control Endpoint • IN transactions with DATA1 PID to complete the status phase for a Control Endpoint, when this bit is 0, but bit Data End (bit 4) is 1 • By a BULK IN or ISO IN or INT IN Endpoint The processor should write into the FIFO only if this bit is cleared. After it has completed writing the data, it should set this bit. The data can be of 0 length. For a Control Endpoint, the processor should write to the FIFO only when bit 6 (TX Packet Requested) is set. The H/W clears this bit after it receives an ACK. If the interrupt is enabled, clearing this bit by the H/W causes an interrupt to the processor
3	Stall Snd	W	The USB H/W sets this bit after a STALL is sent to the Host. The firmware uses this bit when responding to a USB GetStatus Request This bit indicates End of data stage for the Control Endpoint only
2	RX SETUP	W	The USB H/W sets this bit when it receives a valid setup packet from the Host. This bit is used by Control Endpoints only to signal to the processor that the USB H/W has received a valid SETUP packet and that the data portion of the packet is stored in the FIFO. The H/W will clear all other bits in this register and will set the RX SETUP. If the corresponding interrupt is enabled, the processor will be interrupted when the RX SETUP is set. After the completion of reading the data from the FIFO, the firmware should clear this bit
1	RX OUT Packet	W	This bit indicates that the USB H/W has decoded an OUT token and that the data is in the FIFO. The USB H/W sets this bit after it has stored the data of an OUT transaction in the FIFO. When this bit is set, the H/W will NAK all OUT tokens. For Control Endpoints only, bit 7 of this register, Enable Control Write, has to be set for the H/W to accept the OUT data. The USB H/W will not overwrite the data in the FIFO except for an early USB Setup Request. Bit RX OUT Packet is used for the following operations: • Control write transactions by a Control Endpoint • OUT transaction with DATA1 PID to complete the status phase of a controlEndpoint • By a BULK OUT or ISO OUT or INT OUT Endpoint Setting this bit causes an interrupt to the processor if the interrupt is enabled. The firmware clears this bit after the FIFO is read
0	TX Complete	R	The H/W sets this bit to indicate to a Control Endpoint that it has received an ACK handshake from the Host. This bit is used by H/W in a Control Endpoint to signal to the processor that it has successfully completed certain transactions. TX Complete is set at the completion of a: • Control read data stage • Status stage without data stage • Status stage after a control write transaction

ECR (Endpoint Control Registers 0 –7)

addr: See Table 8-1 8 bits

Bit	Field	AVR	Description
7	EPEDS	R	Endpoint Enable/Disable (0 = Disable Endpoint, 1 = Enable Endpoint)
6	Reserved	R	Reserved
5:4	Reserved	Reserved and set to 0	
3	DTGLE	W	Data Toggle. Identifies DATA0 or DATA1 packets
2	EPDIR	R	Endpoint Direction Only applicable for non-Control Endpoints (0 = Out, 1 = In)
1:0	EPTYPE	R	Endpoint Type These bits represent the type of the Endpoint as follows: Bit1 Bit0 Type 0 0 Control 0 1 Isochronous 1 0 Bulk 1 1 Interrupt

ENDPPGPG (Endpoint Ping-Pong Enable Register)

addr: $F0F1 8 bits

Bit	Field	AVR	Description
7	Reserved		
6	PG PG 6 EN	R	Enable Endpoint 6 Ping-pong
5	PG PG 5 EN	R	Enable Endpoint 5 Ping-pong
4	PG PG 4 EN	R	Enable Endpoint 4 Ping-pong
3	PG PG 3 EN	R	Enable Endpoint 3 Ping-pong
2	PG PG 2 EN	R	Enable Endpoint 2 Ping-pong
1	PG PG 1 EN	R	Enable Endpoint 1 Ping-pong
0	PG PG 0 EN	R	Enable Endpoint 0 Ping-Pong

FADDR (Function Address Register)

addr: $F0F2 8 bits

Table 1.

Bit	Field	AVR	Description	
7	FEN	R	Function Enable	The FIU contains an address register that contains the function address assigned by the Host. The Function Address register must be programmed by the processor once it has received a SET_ADDRESS command from the Host and completed the status phase of the transaction. After power up or reset, this register will contain the value of 0x00. The Function Enable bit (FEN) allows the firmware to enable or disable the function Endpoints. The firmware will set this bit after receipt of a reset through the USB H/W. Once this bit is set, the USB H/W passes packets to and from the Host
6:0	FADD[6:0]	R	Function address	

UIER (USB Interrupt Enable Register)

addr: $F0F3 8 bits

Bit	Field	AVR	Description	
7	SOF IE	R	Enable SOF Interrupt	The bits in this register have the following meaning: 1 = Enable interrupt 0 = Disable interrupt
6	EP6 IE	R	Enable Endpoint 6 Interrupt	
5	EP5 IE	R	Enable Endpoint 5 Interrupt	
4	EP4 IE	R	Enable Endpoint 4 Interrupt	
3	EP3 IE	R	Enable Endpoint 3 Interrupt	
2	EP2 IE	R	Enable Endpoint 2 Interrupt	
1	EP1 IE	R	Enable Endpoint 1 Interrupt	
0	EP0 IE	R	Enable Endpoint 0 Interrupt	

UIAR (USB Interrupt Acknowledge Register)

addr: $F0F5 8 bits

Bit	Field	AVR	Description	
7	Reserved			The bits in this register are used to indirectly clear the bits of the UISR. A bit in the UISR is cleared if a 1 is written in the corresponding bit of UIAR
6	EP6 INTA	W	Endpoint 6 Interrupt Acknowledge	
5	EP5 INTA	W	Endpoint 5 Interrupt Acknowledge	
4	EP4 INTA	W	Endpoint 4 Interrupt Acknowledge	
3	EP3 INTA	W	Endpoint 3 Interrupt Acknowledge	
2	EP2 INTA	W	Endpoint 2 Interrupt Acknowledge	
1	EP1 INTA	W	Endpoint 1 Interrupt Acknowledge	
0	EP0 INTA	W	Endpoint 0 Interrupt Acknowledge	

UISR (USB Interrupt Status Register)

addr: $F0F7 8 bits

Bit	Field	AVR	Description	
7	Reserved			The function interrupt bits will be set by the H/W whenever the following bits in the corresponding Endpoint's Control and Status register are modified by the USB H/W: RX OUT Packet is set (Control and OUT Endpoints) TX Packet Ready is cleared (Control and IN Endpoints) RX SETUP is set (Control Endpoints only) TX Complete is set (Control Endpoints only)
6	EP6 INT	W	Endpoint 6 Interrupt	
5	EP5 INT	W	Endpoint 5 Interrupt	
4	EP4 INT	W	Endpoint 4 Interrupt	
3	EP3 INT	W	Endpoint 3 Interrupt	
2	EP2 INT	W	Endpoint 2 Interrupt	
1	EP1 INT	W	Endpoint 1 Interrupt	
0	EP0 INT	W	Endpoint 0 Interrupt	

SPRSIE (Suspend/Resume Interrupt Enable Register)

addr: $F0F9 8 bits

Bit	Field	AVR	Description
7:4	Reserved		
3	SOF IE	R	Enable SOF Interrupt
2	EXTRSM IE	R	Enable External Resume Signaling Interrupt 1 = enable 0 = disable
1	RCVDRSM IE	R	Enable BUS Resume Signaling Interrupt 1 = enable 0 = disable
0	SUSP IE	R	Enable Suspend Signaling Interrupt 1 = enable 0 = disable

SPRSR (Suspend/Resume Register)

addr: $F0FA 8 bits

Bit	Field	AVR	Description
7:4	Reserved		
3	SOF INT	R/W	Start Of Frame Interrupt. Firmware clears this bit to acknowledge the SOF interrupt.
2	EXT RSM	R/W	Received External Resume. The USB H/W sets this bit to denote an External Resume Interrupt. If RMWUPE =1, a RESUME signal is send in USB BUS. Firmware clears this bit to acknowledge the EXT RSM interrupt.
1	RCVD RSM	R/W	Received Resume. The USB H/W sets this bit when a USB resume signaling is detected at its port. Firmware clears this bit to acknowledge the RCVD RSM interrupt.
0	SUSP	R/W	Suspend. The USB H/W sets this bit when it detects no SOF for 3ms. The USB macro enters in SUSPEND MODE, the processor has to go in SLEEP mode.Firmware clears this bit to acknowledge the SUSP interrupt.

GLB_STATE (Global State Register)

addr: $F0FB 8 bits

Bit	Field	AVR	Description
7:4	Reserved		Reserved
3	RSMINPR	W	Set by the H/W when a Resume is send in the USB bus during Remote Wake-up feature (13 ms)

Bit	Field	AVR	Description
2	RMWUPE	R	Remote Wake-up Enable. This bit is set if the Host enables the function's remote wake-up feature
1	CONFG	R	Configured. This bit is set by the firmware after a valid SET_CONFIGURATION request is received. It is cleared by a reset or by a SET_CONFIGURATION with a value of 0
0	FADD Enable	R	Function Address Enable. This bit is set by firmware after the status phase of a SET_ADRESS request transaction. The Host will use the new address starting at the next transaction

FRM_NUM_L (Frame Number Low Register)

addr: $F0FC 8 bits

Bit	Field	AVR	Description
7:0	FCL[7:0]	W	This is the lower 8-bits of the 11 bit frame number of SOF packet

FRM_NUM_H (Frame Number High Register)

addr: $F0FD 8 bits

Bit	Field	AVR	Description
7:3	Reserved		Reserved and set to 0
2:0	FCH[10:8]	W	This is the upper 3 bits of the 11 bit frame number of SOF packet

8.2 UART Register Set

The base address for UART0 registers is $F200 and for UART1 registers is $F300. Each read or write access of the UART registers consumes at least 2 CPU cycles, since the UART core clock is asynchronous and fixed to 14.769 MHz.

The register file and its fields are briefly presented in Table 26. A more detailed description is provided in the following sections.

Table 8-2. UART Register File and Register Fields

Addr A[3:0]	Register	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
General Register Set									
0000 DLAB = 0	RHR	MSB							LSB
00000 DLAB = 0	THR	MSB							LSB

Table 8-2. UART Register File and Register Fields (Continued)

0001 DLAB = 0	IER	0	0	0	0	Enable Modem Status Interrupt	Enable Rx Line Status Interrupt	Enable THR Ready Interrupt	Enable Rx Data Available Interrupt
0010	IIR	FIFO Enabled	FIFO Enabled	0	0	Interrupt ID Bit 2	Interrupt ID Bit 1	Interrupt ID Bit 0	Not Interrupt Pending
0010	FCR	Rx FIFO Trigger MSB	Rx FIFO Trigger LSB	0	0	DMA Mode	Tx FIFO Reset	Rx FIFO Reset	FIFO Enable
0011	LCR	Divisor Latch Access (DLAB)	Set Break	Stick Parity	Even Parity	Enable Parity	Number of Stop Bits	Word Length Bit 1	Word Length Bit 0
0100	MCR	0	0	0	Loopback	OUT2	OUT1	RTS	DTR
0101	LSR	Error in Rx FIFO	Transmitte r Empty	THR Ready	Break Interrupt	Framing Error	Parity Error	Overrun Error	Rx Data Available
0110	MSR	CD	RI	DSR	CTS	Delta CD	Trailing Edge RI	Delta DSR	Delta CTS
0111	SCR	MSB							LSB
Special Register Set									
0000 DLAB = 1	DLL	MSB							LSB
0001 DLAB = 1	DLH	MSB							LSB
1000	XR1	Output pins in Loopback	Send Address	Start Time-out Control	Restart Time-out	Tx Reset	Rx Reset	Tx Disable	Rx Disable
1001	XR2	Tx FIFO Disable	Transmitte r Empty	THR Ready	THR Ready bit Control	0	0	0	Multi-drop
1010	MDR	nCD pin Direction	nRI pin Direction	nDSR pin Direction	nCTS pin Direction	nOut2 pin Direction	nOut1 pin Direction	nRTS pin Direction	nDTR pin Direction
1011	RTO	MSB							LSB
1100	TTG	MSB							LSB

Receive Holding Register (RHR)

addr: $0000 8 bits

Bit	Field	Default	AVR	**Description**
7	MSB	0	R	This bit holds the received byte
6:1		0	R	
0	LSB	0	R	

Transmit Holding Register (THR)

addr: $0000 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	W	This bit holds the byte to be transmitted
6:1		0	W	
0	LSB	0	W	

Interrupt Enable Register (IER)

addr: $0001 8 bits

Bit	Field	Default	AVR	Description
7:4	—	0	R/W	—
3	MSI: Enable Modem Status Interrupt	0	R/W	If this bit is set, it enables the MODEM status interrupt (MSR[7:4])
2	RLSI: Enable Receive Line Status Interrupt	0	R/W	If this bit is set, it enables the line status interrupt (LSR[4:1])
1	TRI: Enable THR Ready Interrupt	0	R/W	If this bit is set, it enables the THR ready interrupt (LSR[5])
0	RDAI: Enable RX Data Available Interrupt	0	R/W	If this bit is set, it enables the Rx data available interrupt. It also enables the time-out interrupt when the time-out counter is enabled (in FIFO mode or if RTO is not 0, see XR1[5] and RTO)

Interrupt Identification Register (IIR)

addr: $0010 8bits

Bit	Field	Default	AVR	Description
7:6	FIFOEN: FIFO Enabled	0	R	These two bits are set when FCR[0] is set
5:4	—	0	R	—
3	ID2	0	R	Interrupt ID Bit 2

Bit	Field	Default	AVR	Description
2	ID1	0	R	Interrupt ID Bit 1
1	ID0	0	R	Interrupt ID Bit 0
0	NIP:Not Interrupt Pending	0	R	When this bit is in logic 0, then an interrupt is pending and the IIR[3..1] bits may be used for interrupt type identification. When this bit is in logic 1, no interrupt is pending

Table 8-3. UART Interrupt Priority

Priority	ID2	ID1	ID0	Interrupt Source (Event)	Interrupt Reset Control
Highest	0	1	1	Receiver Line Status (LSR[3..1] not 0)	Reading LSR
	0	1	0	Received Data Available	Reading all the data from the Receive Holding Register or the Rx FIFO
	1	1	0	Time-out Indication	Reading the RHR or receiving a new word from SIN pin
	0	0	1	Transmitter Holding Register Ready	Reading IIR or writing to THR
Lowest	0	0	0	Modem Status (when any bit in MSR[7..4] changes from '0' to '1').	Reading MSR

FIFO Control Register (FCR)

addr: $0010 8 bits

Bit	Field	Default	AVR	Description
7	RCVR1	0	W	RCVR trigger bits. These bits indicate the minimum number of bytes required in the receive FIFO to generate a receive ready interrupt. The trigger level is shown from the following table
6	RCVR0	0	W	
5:4	—	0	W	—
3	RDMA: DMA Mode select	0	W	When FIFOs are disabled (FCR[0] is low), this bit is forced to 0. When set to logic 1, the DMA is in burst mode allowing transfers until the Rx FIFO has been emitted or the Tx FIFO has been filled. When it is cleared to 0, the DMA is in single mode and the words are read one word at a time

Bit 7	Bit 6	Trigger Level (Words)
0	0	1
0	1	4
1	0	8
1	1	14

Bit	Field	Default	AVR	Description
2	TRS:Tx FIFO reset	0	W	When this bit is set, it resets the transmit FIFO
1	FRS:Rx FIFO reset	0	W	When this bit is set, it resets the receive FIFO
0	FEN:FIFO enable	0	W	When this bit is set, it enables the 16-byte receive and transmit FIFOs. When this bit is cleared, the FIFOs are disabled and reset

Line Control Register (LCR)

addr: $0011 8 bits

Bit	Field	Default	AVR	Description
7	DLAB: Divisor Latch Access	0	R/W	This bit must be set to logic 1 during a read or write operation in order to access the Divisor Latches. Resetting this bit to 0 allows access to RHR, THR, and IER
6	SBRK: Set Break	0	R/W	If this bit is set, then it causes a break condition to to be transmitted to the receiving UART. The SOUT pin is forced to the spacing state (logic 0). Resetting to logic 0 stops the break condition. The break control bit acts only on SOUT pin and has no effect on the transmitter logic. Note that UART waits before starting the break condition command for the complete transmission of the word in the transmit shift register. There is no need for software synchronization
5	SPAR: Stick Parity	0	R/W	When the Parity Enable and the Parity Stick bits are set to logic 1, then the Even Parity bit controls the transmitted parity value. By resetting the Even Parity bit to logic 0, the parity bit is transmitted and checked as 1. By setting the Even Parity bit to logic 1, the parity bit is transmitted and checked as 0
4	EVPAR: Even Parity	0	R/W	When the Enable Parity bit is a 1 and the Stick Parity bit is a 0, then by setting to the Even Parity bit o 1 an even number of logic 1s is transmitted or checked in the data word bits and the Parity bit. When Even Parity bit is reset to logic 0, an odd number of 1s are transmitted or checked
3	ENPAR: Enable Parity	0	R/W	By setting this bit to logic 1, a parity bit is transmitted or checked. Resetting this bit to 0, no parity bit is transmitted or checked

Bit	Field	Default	AVR	Description
2	SB: Number of Stop Bits	0	R/W	This bit determines the number of stop bits according to the following table: Bit 2 / Word Length / Number of Stop Bits 0 / any / 1 1 / 5 / 1,5 1 / 6, 7, 8 / 2
1	WL1: Word Length	0	R/W	These bits determine the word length according to the following table Bit 1 / Bit 0 / Word Length (Bits) 0 / 0 / 5 0 / 1 / 6 1 / 0 / 7 1 / 1 / 8
0	WL0: Word Length	0	R/W	

Modem Control Register (MCR)

addr: $0100 8 bits

Bit	Field	Default	AVR	Description
7:5	—	0	R/W	
4	LB: Loopback	0	R/W	If this bit is set, it enables the loopback mode. This bit cannot be set to logic 1 if the value of MDR is not $F0
3	OUT2	0	R/W	The compliment value of the bi-directional pin nOUT2
2	OUT1	0	R/W	The compliment value of the bi-directional pin nOUT1
1	RTS	0	R/W	The compliment value of the bi-directional pin nRTS
0	DTR	0	R/W	The compliment value of the bi-directional pin nDTR

Line Status Register (LSR)

addr: $0101 8 bits

Bit	Field	Default	AVR	Description
7	ERF: Error in RX FIFO	0	R	If the FIFOs are disabled, this bit is a 0. If the FIFOs are enabled, this bit indicates that at least one word in the Rx FIFO has its Parity Error, Framing Error, or Break Indication bits high
6	TE: Transmitter Empty	0	R	If set, this bit indicates that both the Transmit Shift register and Transmit Holding register, or the Tx FIFO if Tx FIFO is enabled, are empty
5	THRR: Transmit Holding Register ready	0	R	If set, this bit indicates that the THR is ready to accept a new word for transmission. This bit is set when a word is transferred from the THR into the Tx Shift register. This bit is reset concurrently with the loading of the THR by the core. If the Tx FIFO is enabled (FCR[0] = 1, XR2[7] = 0), the function of this bit is controlled by XR2[4]. If XR2[4] is 0, then this bit is set when the Tx FIFO is empty; it is cleared when at least 1 word is written to the Tx FIFO. If XR2[4] is 1, then this bit is set when the Tx FIFO is not full
4	BI: Receive Break Interrupt	0	R	If set, this bit indicates a Break Interrupt that the receive data input is held in the spacing state (logic 0) for longer than a full word transmission time.In FIFO mode, this error is associated with the word at the top of the Rx FIFO which is equivalent to RHR. This bit is reset to a logic 0 whenever the core reads the LSR
3	FE: Framing Error	0	R	If set, this bit indicates a framing error. The received word in RHR does not have the correct stop bit. In FIFO mode, this error is associated with the word at the top of the Rx FIFO which is equivalent to RHR. This bit is reset to a logic 0 whenever the core reads the LSR
2	PE:Parity Error	0	R	If set, this bit indicates a parity error. The received word in RHR does not have the correct parity bit. In FIFO mode, this error is associated with the word at the top of the FIFO which is equivalent to RHR. This bit is reset to a logic 0 whenever the core reads the LSR
1	OE:Overrun Error	0	R	If set, this bit indicates an overrun error, that is, the data in RHR was not read by the core before the next word was transferred into the RHR, thereby destroying the previous word. In FIFO mode, an overrun error will occur only after the Rx FIFO is full and the next word has been completely received in the Shift register. The word in the shift register is overwritten, but it is not transferred to the Rx FIFO. The overrun error is indicated to the core as soon as it happens. This bit is reset to a logic 0 whenever the core reads the LSR
0	RDA: Receive Holding Register Ready	0	R	If set, this bit indicates that there is data available in the RHR. This bit resets to logic 0 by reading all of the data from the Receive Holding Register or the Rx FIFO

Modem Status Register (MSR)

addr: $0110 8 bits

Bit	Field	Default	AVR	Description
7	CD	0	R/W	The compliment of the bi-directional Carrier Detect nCD I/O pin. If MCR[4] is set (loopback mode), then this bit is equivalent to nOUT2 pin
6	RI	0	R/W	The compliment of the bi-directional Ring Indicator nRI I/O pin. If MCR[4] is set (loopback mode),then this bit is equivalent to nOUT1 pin
5	DSR	0	R/W	The compliment of the bi-directional Data Set Ready nDSR I/O pin. If MCR[4] is set (loopback mode), then this bit is equivalent to nDTR pin
4	CTS	0	R/W	The compliment of the bi-directional Clear To Send nCTS I/O pin. If MCR[4] is set (loopback mode), then this bit is equivalent to nRTS pin
3	DCD: Delta Carrier Detect indicator	0	R/W	This bit indicates that the nCD pin has changed state since the last time it was read by the core
2	TRI: Trailing Edge of Ring Indicator	0	R/W	This bit indicates that the nRI pin has changed from a low to a high state since the last time it was read by the core
1	DDSR: Delta Data Set Ready indicator	0	R/W	This bit indicates that the nDSR pin has changed states since the last time it was read by the core
0	DCTS: Delta Clear To Send indicator	0	R/W	This bit indicates that the nCTS pin has changed states since the last time it was read by the core

Scratch-pad Register (SCR)

addr: $0111 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	A scratch-pad register which holds data temporarily. Doe not effect the UART
6:1		0	R/W	
0	LSB	0	R/W	

Divisor Latch Register, Low Byte (DLL)

addr: $0000 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	Baud rate generator division ratio low byte
6:1		0	R/W	
0	LSB	0	R/W	

Divisor Latch Register, High Byte (DLH)

addr: $0001 8 bits

Bit	Field	Default	AVR	Description
7	MSB	0	R/W	Baud rate generator division ratio high byte The main UART clock, from the system clock generator, is divided by the 16-bit number contained in DLL and DLH, to provide the UART clock (which is 16 times the actual serial data rate)
6:1		0	R/W	
0	LSB	0	R/W	

Extra Register 1 (XR1)

addr: $1000 8 bits

Bit	Field	Default	AVR	Description
7	OPL:Output pins in Loopback	0	R/W	If this bit is set in loopback mode (MCR[4] is set), then the SOUT, nOUT2, nOUT1, nRTS, and nDTR pins are not forced to logic 1. Instead, these pins operate normally (echo). If the loopback mode is disabled, then this bit has no effect
6	SA:Send Address	0	R/W	Effective in multidrop mode only (see XR2[0]). If set, the next word will be transmitted with the parity bit (in multidrop mode, this is called address bit) forced at logic 1
5	STOC: Start Time-out Control	0	R/W	This bit controls the start operation of the time-out counter. If set in logic 0, then it operates in 16550 compatible mode. If set in logic 1, the time-out counter starts whenever RTO is not 0 (see and RTO)
4	RTO: Restart Time-out	0	R/W	Writing a logic 1 resets the time-out counter. This bit resets to logic 0 automatically
3	TxR: Tx Reset	0	R/W	Writing a logic 1 resets the transmit path (and Tx FIFO). This bit resets to logic 0 automatically
2	RxR: Rx Reset	0	R/W	Writing a logic 1 resets the receive path (and Rx FIFO). This bit resets to logic 0 automatically
1	TxDis: Tx Disable	0	R/W	If this bit is set, it disables the transmit path
0	RxDis: Rx Disable	0	R/W	If this bit is set, it disables the receive path

Note: When XR1 and XR2 registers are both $00, then the UART operates in 16550 compatible mode.

Extra Register 2 (XR2)

addr: $1001 8 bits

Bit	Field	Default	AVR	Description
7	TxFD:Tx FIFO Disable	0	R/W	If this bit is set, it disables the Tx FIFO. So, if FCR[0] and XR2[7] are both set, then only the Rx FIFO is enabled
6	TE:Transmitter Empty	0	R/W	This bit is equivalent to LSR[6]. It can be used to check if the transmitter is empty without resetting the error bits in LSR
5	THRR: THR Ready	0	R/W	This bit is equivalent to LSR[5]. It can be used to check if the THR is ready to load data without resetting the line status bits in LSR
4	THRRC:THR Ready Bit Control	0	R/W	Functional only if Tx FIFO is enabled. If set in logic 0, then the LSR[5] (and XR2[5]) bit indicates that Tx FIFO is empty (THRR bit is 1) or that it has at least one word waiting for transmission (THRR bit is 0). Setting this bit to logic 1, then LSR[5] (and XR2[5]) bit indicates that Tx FIFO is not full (THRR bit is 1) or that it is full (THRR bit is 0)
3:1	—	0	R/W	—
0	MDM: Multidrop Mode	0	R/W	If set, this bit enables the multidrop mode. In this case, the Parity Error Bit in LSR is set when data is detected with the parity bit at logic 1 to identify an address word. If the received parity bit is detected low, then the Parity Error bit is not set. The transmitter sends an address word (with the parity bit set) when the Send Address bit (XR1[6]) is set. Setting the XR1[6] the next word written to THR will be transmitted as an address and any transmitted word after this will have the parity bit cleared

Note: When XR1 and XR2 registers are both $00, then the UART operates in 16550 compatible mode.

Modem Direction Register (MDR)

addr: $1010 8 bits

Bit	Field	Default	AVR	**Description**
7	nCDD:nCD pin Direction	1	R/W	If this bit is set, the nCD pin is configured as input
6	nRID:nRI pin Direction	1	R/W	If this bit is set, the nRI pin is configured as input
5	nDSRD:nDSR pin Direction	1	R/W	If this bit is set, the t nDSR pin is configured as input
4	nCTSD:nCTS pin Direction	1	R/W	If this bit is set, the nCTS pin is configured as input
3	nOUT2D:nOUT2 pin Direction	0	R/W	If this bit is set, the nOUT2 pin is configured as input

Bit	Field	Default	AVR	**Description**
2	nOUT1D:nOUT1 pin Direction	0	R/W	If this bit is set. the nOUT1 pin is configured as input
1	nRTSD:nRTS pin Direction	0	R/W	If this bit is set, the nRTS pin is configured as input
0	nDTRD:nDTR pin Direction	0	R/W	If this bit is set, the nDTR pin is configured as input

Receiver Time-out Register (RTO)

addr: $1011 8 bits

Bit	**Field**	**Default**	**AVR**	**Description**
7	MSB	0	R/W	This register contains the maximum bit periods, for which the UART will wait the next word to arrive. Whenever the time-out counter expires, then a Time-out Indication Interrupt will be issued The XR1[5] Start Time-out Control bit selects the Start Time-out and RTO load mechanism. If the XR1[5] bit is reset to 0 (16550 compatible mode), then the RTO loads the value 4 times the word length + 12 on each LCR write operation. After reset, the word length is 5 bits and the RTO is $20. The time-out counter will then start counting down only in FIFO mode (FCR[0] is set) and if the Rx FIFO holds at least 1 word. If XR1[5] is set to 1, then the time-out function is available and in FIFO disabled mode. The RTO value does not change with LCR write operations. The time-out counter will start counting down whenever the RTO is not $00 and the RHR is loaded. In all cases core has immediate access at the contents of the RTO. The Time-out counter resets when a new word is completely received and transferred at the Receive Holding register or when the XR1[4] bit is forced to logic 1. If XR1[5] is reset to 0, time-out counter resets and on a RHR read access from core
6		0	R/W	
5		1	R/W	
4		0	R/W	
3		0	R/W	
2		0	R/W	
1		0	R/W	
0	LSB	0	R/W	

Transmitter Time Guard Register (TTG)

addr: $1100 8 bits

Bit	**Field**	**Default**	**AVR**	**Description**
7	MSB	0	R/W	The value of this register indicates the delay (in bit periods) that an active transmitter has to interpose between two consecutive word transmissions
6:1		0	R/W	
0	LSB	0	R/W	

Table 8-4. Baud Rate Generation Example (UART Clock = 14,769 MHz)

Output Baud Rate	User Devisor (16*clk)		UBM Value	UBL Value
	Decimal	Hex	Hex	Hex
100	9216	2400	24	00
200	4608	1200	12	00
400	2304	900	09	00
600	1536	600	06	00
1200	768	300	03	00
2400	384	180	01	80
4800	192	C0	00	C0
9600	96	60	00	60
14400	64	40	00	40
19200	48	30	00	30
28800	32	20	00	20
38400	24	18	00	18
57600	16	10	00	10
76800	12	0C	00	0C
115200	8	08	00	08
153600	6	06	00	06
230400	4	4	00	04
307200	3	03	00	03
460800	2	02	00	02
921600	1	01	00	01

8.3 Memory Access Interface Register Set

Memory Bank Map Register (MEMMAP)

addr: $F800 8 bits

Bit	Field	Default	AVR	**Description**
7	MSB	0	R/W	Utilizing the MEMMAP register, it is possible to resize the two memory spaces
6:1		0	R/W	
0	LSB	1	R/W	

DMA External Memory Interface Control Register A (DMA_EMICRA)

addr: $F801 8 bits

Bit	Field	Default	AVR	Description
7	RW1	0	R/W	Read Wait States: These bits control the wait states inserted in the corresponding (read, write, and ALE) signals
6	RW0	0	R/W	
5	RM1	0	R/W	Read Mode Select: These bits control the mode (waveform) of the corresponding (read, write, and ALE) signals
4	RM0	0	R/W	
3	WW1	0	R/W	Write Wait States:These bits control the wait states inserted in the corresponding (read, write, and ALE) signals
2	WW0	0	R/W	
1	WM1	0	R/W	Write Mode Select: These bits control the mode (waveform) of the corresponding (read, write, and ALE) signals
0	WM0	0	R/W	

DMA External Memory Interface Control Register B (DMA_EMICRB)

addr: $F802 8 bits

Bit	Field	Default	AVR	Description
7	AW1	0	R/W	ALE Wait States: These bits control the wait states inserted in the corresponding (read, write, and ALE) signals
6	AW0	0	R/W	
5	AM1	0	R/W	ALE Mode Select: These bits control the mode (waveform) of the corresponding (read, write, and ALE) signals
4	AM0	0	R/W	
3	—	0	R/W	These bits are reserved and must be remain always in 0 value
2	—	0	R/W	
1	EMD0	0	R/W	External Memory Device Select: These bits select the external memory interface mode according to the following table:
0	EMD1	0	R/W	

EMD1	EMD0	External Memory Interface Mode
0	0	FIFO
0	1	Reserved
1	0	Demultiplexed
1	1	Multiplexed

9. Errata

1. Stack Pointer is 11-bits wide

The Stack Pointer is 11-bits wide.

Problem Fix/Workaround

Keep the stack below the address $07FF.

2. DDRB initial value

The boostrap ROM code, after accessing the external SPI memory and remmaping to the final code, leaves the DDRB register to $10 value (instead of $00). The result is to have the “PB4/nSS” pin as output and set to 0 value (because PORTB = $00).

Problem fix/workaround

Set the correct value of the DDRB register according to the given system configuration.

3.USB “setup” packet

In some cases the USB controller might not respond to a “setup” packet if the previous USB packet was a “status-in” and the firmware performed an action while servicing the packet.

Problem fix/workaround

The firmware must wait for the next “SOF” packet before servicing the “status-in” packet. The “Set Address” is the only USB standard request that needs this special treatment.

4. IIR and MSR UART registers

The IIR Register indicates when any bit in MSR[7..4] is changed from 0 to 1 but it does not indicate a transition from 1 to 0.

Problem fix/workaround

The firmware needs to poll the MSR register in order to detect if any bit in MSR[7..4] changes its value from 1 to 0.

5.Unserved UART Tx Interrupt Request

The problem arises when both Receive (Rx) and Transmit (Tx) IRQs are enabled and both Rx and Tx IRQs are pending. In that case the firmware detects the Rx IRQ when it reads the IIR. This IIR read though, may erroneously clear the Tx IRQ. Thus, the firmware loses the Tx IRQ, which is never served.

Problem Fix / Workaround

The UART Interrupt Service Routine (ISR) must examine and serve a potential Tx IRQ, independently from the IIR value. For example, the ISR can use either the LSR[5] or the XR2[5] bits to check for pending Tx IRQs.

10. Electrical Specifications

10.1 Absolute Maximum Ratings

Operating Temperature .....-40°C to 85°C
Storage Temperature .....-65°C to 150°C
Voltage on Any Pin with Respect to Ground ..... 0V to Maximum Operating Voltage
Maximum Operating Voltage 3.6V

*NOTICE: Stresses beyond those listed under “Absolute Maximum Ratings” may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

10.2 D.C. Characteristics

10.2.1 Power Supply

Symbol	Parameter	Condition	Min	Type	Max	Unit
CVDD	Core Supply Voltage		1.65	1.8	1.95	V
PVDD	Periphery Supply Voltage		CVDD	3.3	CVDD + 1.5, 3.6 max	V
PVDD_USB	USB Supply Voltage		3.0	3.3	3.6	V
PAVDD	Analog Power Supply		1.65	1.8	1.95	V
V_{IL}	Low Level Input Voltage		-0.3		+0.8	V
V_{IH}	High Level Input Voltage		2.0		PVDD + 0.3	V
V_{OL}	Low Level Output Voltage	I_{OL} = 0mA			0.2	V
		I_{OL} = 2mA			0.4	V
V_{OH}	High Level Output Voltage	I_{OH} = 0mA	PVDD - 0.2			V
		I_{OH} = 2mA	PVDD - 0.4			V

10.2.2 Pull-up Circuit (where pin = 0V)

	Condition	Min	Max
Current	PVDD = 3.3V	129µA (PVDD = 3.0V)	322µA (PVDD = 3.6V)
Current	PVDD = 1.8V	30µA (PVDD = 1.65V)	92µA (PVDD = 1.95V)

10.2.3 Pull-down Circuit (where pin = PVDD)

	Condition	Min	Max
Current	PVDD = 3.3V	110µA (PVDD = 3.0V)	356µA (PVDD = 3.6V)
Current	PVDD = 1.8V	27µA (PVDD = 1.65V)	106µA (PVDD = 1.95V)

10.2.4 USB Signals: DP, DM

Code	Parameter	pu33b11f	Unit
VT+	High Level Input Voltage	1.8	V
VT-	Low Level Input Voltage	0.8	V
Vhys	Hysteresis	0.2	V

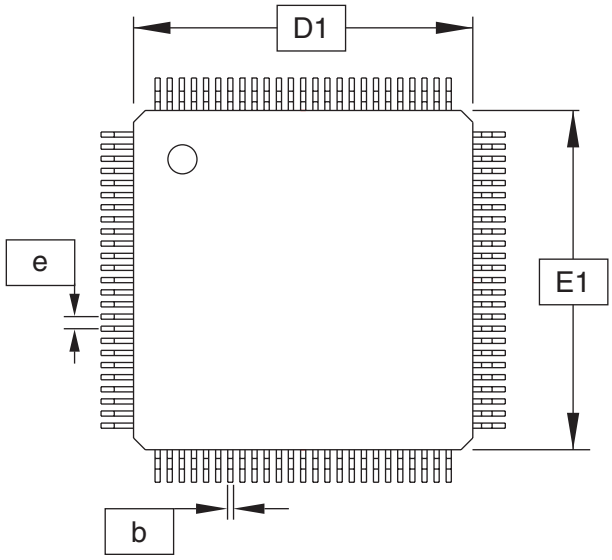
10.2.5 Oscillator Signals: OSC

Symbol	Parameter	Condition	Min	Typ	Max	Unit
PAVDD	Supply Voltage	See Standard Operating Conditions Below	1.65	1.8	1.95	V
ΔV_{dd}	Supply Ripple	rms value, 10 KHz to 10 MHz			30	mV
idd on	Current Consumption	@16 MHz		0.9	1.6	MA
Freq	Operating Frequency		8	12	16	MHz
Duty	Duty Cycle		40		60	%
Ton	Startup Time	With Crystal Defined Below			2	ms
Pon	Drive Level				150	µW
ESR	Equivalent Serie Resistance	@16 MHz			80	Ω
Cm	Motional Capacitance		5		9	fF
Cshunt	Shunt Capacitance				7	pF
Cload	Load Capacitance	Max. External Capacitors:40pF	15		20	pF
Idd stdby	Standby Current Consumption	onosc = 0			1	µA

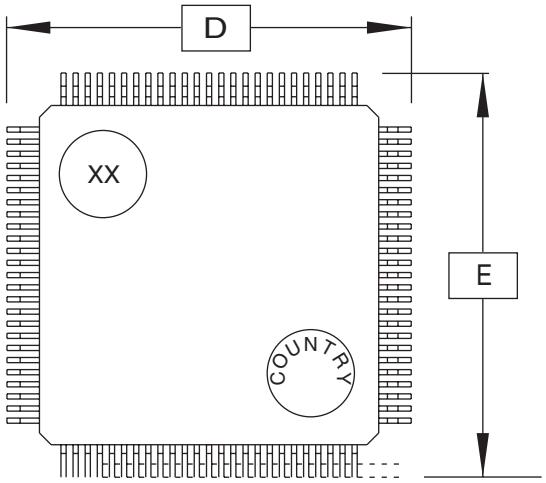
10.3 Ordering Information

Device	Ordering Code	Package
AT76C713	AT76C713	100 lead TQFP

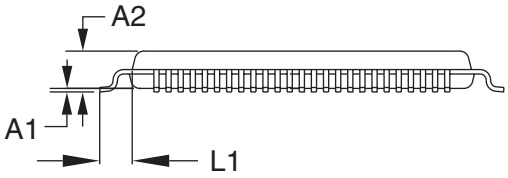
11. Packaging Information



Top View



Bottom View



Side View

COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A1	0.05		0.15	6
A2	0.95	1.00	1.05	
D		16.00 BSC		
D1		14.00 BSC		2, 3
E		16.00 BSC		
E1		14.00 BSC		2, 3
e		0.50 BSC		
b	0.17	0.22	0.27	4, 5
L1		1.00 REF		

Notes:
1. This drawing is for general information only. Refer to JEDEC Drawing MO-153, Variation AA, for proper dimensions, tolerances, datums, etc.
2. The top package body size may be smaller than the bottom package size by as much as 0.15 mm.
3. Dimensions D1 and E1 do not include mold protrusions. Allowable protrusion is 0.25 mm per side. D1 and E1 are maximum plastic body size dimensions, including mold mismatch.
4. Dimension b does not include Dambar protrusion. Allowable Dambar protrusion shall not cause the lead width to exceed the maximum b dimension by more than 0.08 mm. Dambar cannot be located on the lower radius or the foot. Minimum space between protrusion and an adjacent lead is 0.07 mm for 0.4 and 0.5 mm pitch packages.
5. These dimensions apply to the flat section of the lead between 0.10 mm and 0.25 mm from the lead tip.
6. A1 is defined as the distance from the seating place to the lowest point on the package body.

11/30/01

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**100T1**, 100-lead (14 x 14 x 1.0 mm Body), Thin Plastic Quad Flat Pack (TQFP)	100T1	A

Table of Contents

Atmel Corporation

2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 487-2600

Regional Headquarters

Europe
Atmel Sarl
Route des Arsenaux 41
Case Postale 80
CH-1705 Fribourg
Switzerland
Tel: (41) 26-426-5555
Fax: (41) 26-426-5500

Asia
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimshatsui
East Kowloon
Hong Kong
Tel: (852) 2721-9778
Fax: (852) 2722-1369

Japan
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
Tel: (81) 3-3523-3551
Fax: (81) 3-3523-7581

Atmel Operations

Memory
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

Microcontrollers
2325 Orchard Parkway
San Jose, CA 95131, USA
Tel: 1(408) 441-0311
Fax: 1(408) 436-4314

La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
Tel: (33) 2-40-18-18-18
Fax: (33) 2-40-18-19-60

ASIC/ASSP/Smart Cards
Zone Industrielle
13106 Rousset Cedex, France
Tel: (33) 4-42-53-60-00
Fax: (33) 4-42-53-60-01

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
Tel: (44) 1355-803-000
Fax: (44) 1355-242-743

RF/Automotive
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
Tel: (49) 71-31-67-0
Fax: (49) 71-31-67-2340

1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906, USA
Tel: 1(719) 576-3300
Fax: 1(719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
Tel: (33) 4-76-58-30-00
Fax: (33) 4-76-58-34-80

Literature Requests
www.atmel.com/literature

5665B–USB–04/05