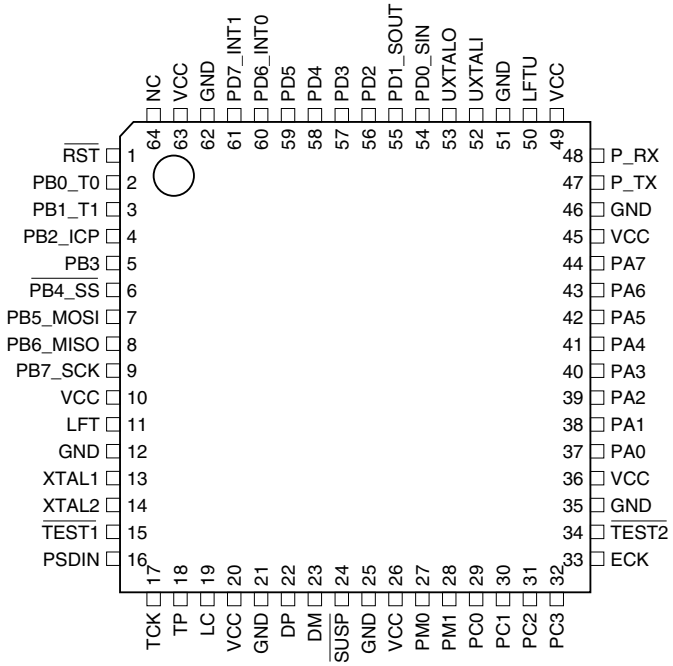
Features

- AVR® Microcontroller
- Clock Generator Provides CPU Rates up to 24 MHz
- Programmable UART with 16-byte FIFOs at the Receiver Side (1), with a Maximum Rate of 921K Baud
- Programmable SPI Interface
- Full-speed USB Function Controller
- On-chip 2K Bytes of SRAM (for Data)
- On-chip 2K Bytes of Dual-port RAM for Segmentation and Reassembly of Packets Exchanged between the USB and the UART Interfaces
- 8K x 16-bit In-System SRAM for Program Code
- On-chip Bootstrap ROM for Program Uploading to the Internal Program SRAM, Either from the USB or the SPI
- One USB Control Endpoint
- Five USB Programmable Endpoints (up to 64 Bytes) with Double-buffered FIFOs for Back-to-back Transfers
- One 8-bit Timer/Counter
- One 16-bit Timer/Counter
- External and Internal Interrupt Sources
- Programmable Watchdog Timer
- Independent UART BRG Oscillator
- 64-lead TQFP Package
- 3.3V Operation

Pin Configuration



AVR®-based Bridge between Full-speed USB and Fast Serial Asynchronous Interfaces

AT76C711

Rev. 1643B–09/01

Description

The Atmel AT76C711 is a compound USB device designed to provide a high-speed USB interface to devices that need to communicate with a base station through fast serial links, like UARTs and IrDA interfaces. It is based on the AVR-enhanced RISC architecture and consists of a USB function interface with a devoted DMA controller

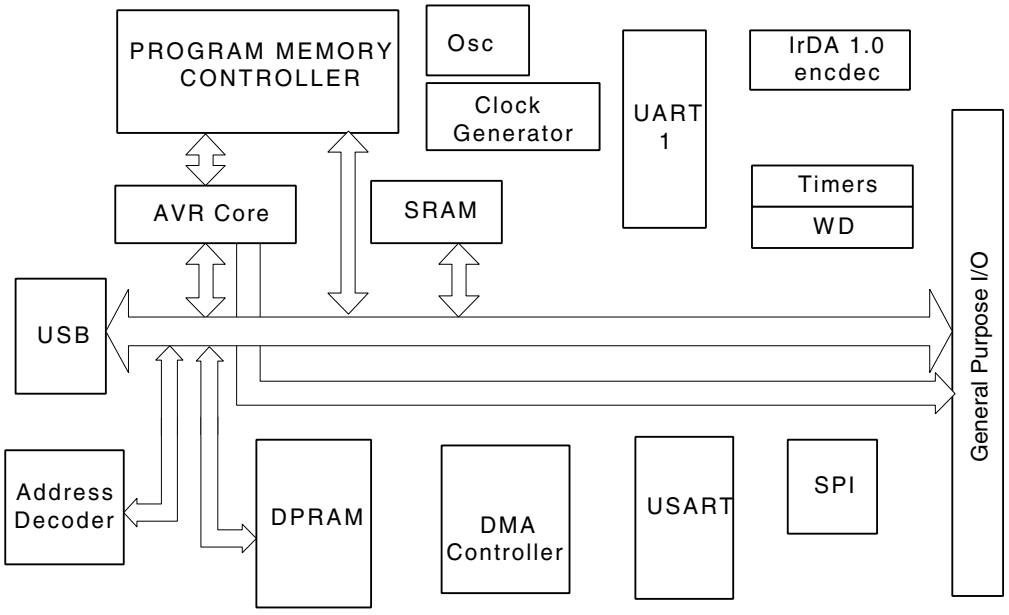
for fast transfers of data between the endpoint FIFOs and the DPRAM, a 2K bytes of internal RAM, a Synchronous Peripheral Interface (SPI), a UART, supporting a maximum rate of 921K baud, an 8K x 16-bit in-system SRAM for microcode storage, which is loaded from the SPI controller, 2K bytes dual-port RAM (DPRAM) and a programmable DMA controller for packet transfers between the UART and the DPRAM, without microcontroller intervention. An IrDA controller is also provided, attached to a second UART module and is able to communicate with an IrDA transceiver with a maximum rate of 1.2 Mbps. A hardwired Device Firmware Upgrade (DFU) protocol handler is implemented for programming an external AT45BDxxx serial Flash during the production phase. An internal bootstrap ROM contains the executable program for uploading the application code from an external serial Flash to the on-chip program SRAM. Alternatively, microcode can be stored in the program SRAM using the slave program mode while the chip is in the reset state. The USB and peripheral device controller function should be implemented in the microcontroller's firmware.

The device is suitable for applications where minimization of power dissipation is required, since there are no power-consumable transactions with external parallel devices.

The USB Hardware block consists of a USB transceiver, the SIE, endpoint controllers and an interface to the microcontroller. The USB Hardware interfaces to the USB host at the packet level. The microcontroller firmware handles the higher-level USB protocol layers that are not processed by the USB Hardware and in addition, it performs the peripheral control functions.

Block Diagram

Figure 1. Block Diagram



Applications

AT76C711 can be used in applications where peripherals supporting fast serial asynchronous or synchronous transfer of data have to communicate with a host or other peripherals through a high-speed serial link, like USB. A typical application of AT76C711 and its functional diagram are shown in Figures 2 and 3.

Typical areas of AT76C711 usage are:

- Connection of Network Interface Cards (NICs) to a host system
- Wireless communications
- Bridging of microcontrollers with different types of serial interfaces
- USB to UART bridge
- USB to IrDA bridge
- IrDA to UART bridge
- Packet adaptation of network protocol packets to USB requirements

Figure 2. Typical AT76C711 Application

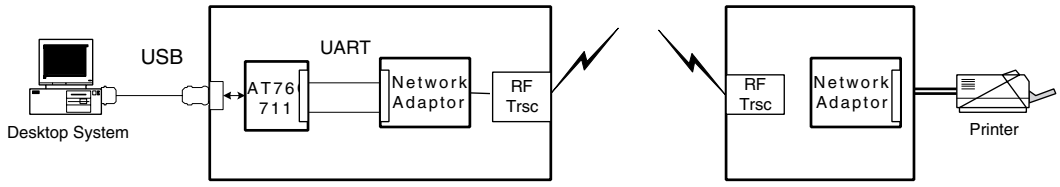
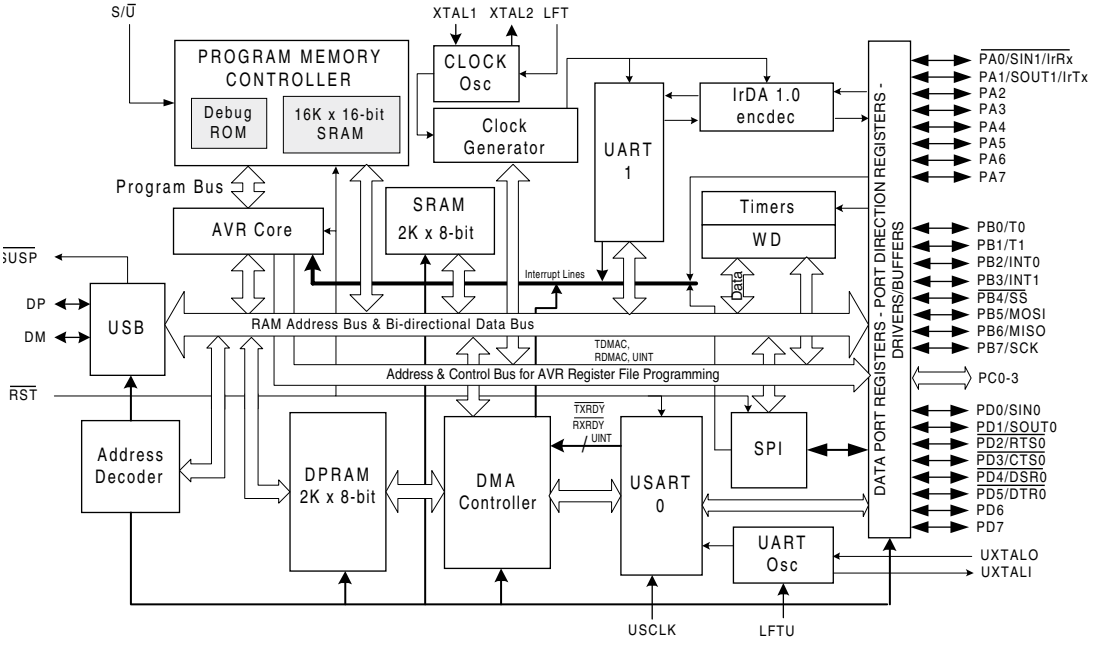


Figure 3. Functional Diagram



Pin Summary – Pin Assignment in Alphabetical Order

Type: I = Input, O = Output, OD = Output, Open Drain, B = Bi-directional, V = Power Supply, Ground

Pin #	Signal	Type
23	DM	B
22	DP	B
33	ECK	I
19	LC	I
14	LFT	I
50	LFTU	I
*	GND	V
64	NC	
48	P_IRX	I
47	P_ITX	O
37	PA0	B
38	PA1	B
39	PA2	B
40	PA3	B
41	PA4	B
42	PA5	B
43	PA6	B

Pin #	Signal	Type
44	PA7	B
2	PB0_T0	B
3	PB1_T1	B
4	PB2_ICP	B
5	PB3	B
6	$\overline{\text{PB4\_SS}}$	B
7	PB5_MOSI	B
8	PB6_MISO	B
9	PB7_SCK	B
29	PC0	B
30	PC1	B
31	PC2	B
32	PC3	B
54	PD0_SIN	B
55	PD1_SOUT	B
56	PD2	B
57	PD3	B

Pin #	Signal	Type
58	PD4	B
59	PD5	B
60	PD6_INT0	B
61	PD7_INT1	B
16	PSDIN	I
1	$\overline{\text{RST}}$	I
27, 28	PM0, PM1	I
24	$\overline{\text{SUSP}}$	O
17	TCK	I
15	$\overline{\text{TEST1}}$	I
34	$\overline{\text{TEST2}}$	I
18	TP	I
**	VCC	V
51	UXTALI	I
52	UXTALO	O
11	XTAL1	I
12	XTAL2	O

Notes: 1. (*) GND: 12, 21, 25, 35, 46, 51, 62.
2. (**) VCC: 10, 20, 24, 36, 45, 49, 63.

Pin Summary – Pin Assignment in Numerical Order

Pin #	Signal	Type
1	$\overline{\text{RST}}$	I
2	PB0_T0	B
3	PB1_T1	B
4	PB2_ICP	B
5	PB3	B
6	$\overline{\text{PB4\_SS}}$	B
7	PB5_MOSI	B
8	PB6_MISO	B
9	PB7_SCK	B
10	VCC	V
11	LFT	I
12	GND	V
13	XTAL1	I
14	XTAL2	O
15	$\overline{\text{TEST1}}$	I
16	PSDIN	I
17	TCK	I
18	TP	I
19	LC	I
20	VCC	V
21	GND	V
22	DP	B

Pin #	Signal	Type
23	DM	B
24	$\overline{\text{SUSP}}$	O
25	GND	V
26	VCC	V
27	PM0	I
28	PM1	I
29	PC0	B
30	PC1	B
31	PC2	B
32	PC3	B
33	ECK	I
34	$\overline{\text{TEST2}}$	I
35	GND	V
36	VCC	V
37	PA0	B
38	PA1	B
39	PA2	B
40	PA3	B
41	PA4	B
42	PA5	B
43	PA6	B
44	PA7	B

Pin #	Signal	Type
45	VCC	V
46	GND	V
47	P_ITX	O
48	P_IRX	I
49	VCC	V
50	LFTU	I
51	GND	V
52	UXTALI	I
53	UXTALO	O
54	PD0_SIN	B
55	PD1_SOUT	B
56	PD2	B
57	PD3	B
58	PD4	B
59	PD5	B
60	PD6_INT0	B
61	PD7_INT1	B
62	GND	V
63	VCC	V
64	NC	V

Signal Description [1]

Type: I = Input, O = Output, OD = Output, Open Drain, B = Bi-directional, V = Power Supply, Ground

Name	Type	Description
Program Memory Controller Signals		
PM0, PM1	I	See Figure 4
PSDIN	I	Program Serial Data-In: In slave program mode, this signal carries the serial program data that are samples with the positive edge of TCK.
TP	I	When $\overline{RST}$ is active (low), a high level of this signal, for at least two TCK pulses, forces the program address generator.
LC	I	Load Complete: A transition from low to high denotes the completion of program data transfer from the external device. The AVR will start executing instructions from the internal SRAM as soon as the $\overline{RST}$ goes high.
TCK	I	A clock signal for sampling PSDIN input.
Port Signals		
PA[0:7]	B	Port A, PB0 through PB7. 8-bit bi-directional port.
PA[0:7]	B	Port B, PB0 through PB7. 8-bit bi-directional port. PB0, PB1, PB2, PB4 through PB7 are dual-function as shown below: **Port** **Alternate Function** PB0 Timer/Counter0 clock input PB1 Timer/Counter1 clock input PB2 (ICP) Input Capture Pin for Timer/Counter1 PB4 (SS#) 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
PC[0:3]	B	Port C, PC0 through PC3. 4-bit output port.
PD[0:7]	B	Port D, PD0 through PD7. 8-bit bi-directional I/O port. PD0, PD1 also serve as the data lines for the asynchronous serial port as listed below: **Port** **Alternate Function** PD0 (SIN) Serial Data-In (I): This pin provides the serial receive data input to 16550 UART. The SIN signal will be a logic “1” during reset, idle (no data). During the local loopback mode, the SIN input pin is disabled and SOUT data is internally connected to the UART SIN input. PD1 (SOUT) Serial Data Out (O): This pin provides the serial transmit data from the 16550 UART. The SOUT signal will be a logic “1” during reset, idle (no data). PD6 (INT0) External Interrupt0 source PD7 (INT1) External Interrupt1 source
USB Serial Interface		
DP	B	Upstream Plus USB I/O. DP and DM form the differential signal pin pair connected to the host controller or an upstream hub.
DM	B	Upstream Minus USB I/O
$\overline{SUSP}$	O	Suspend. This output pin is deactivated (high) during normal operation. It is used to signal the host microcontroller that AT76C711 has received USB suspend signaling. This pin will stay asserted while AT76C711 is in the suspend mode. This pin is deactivated whenever a USB resume signaling is detected on DP and DM.

Signal Description (Continued)[1]

Type: I = Input, O = Output, OD = Output, Open Drain, B = Bi-directional, V = Power Supply, Ground

Name	Type	Description
Test Signals		
$\overline{\text{TEST1}}$	I	Test signal for clocks (used in production phase only – normally tied to high)
$\overline{\text{TEST2}}$	I	Test signal for monitoring internal signal levels using the four data ports (used in production phase only – normally tied to high)
ECK	I	Clock pulse for activating various test modes when $\overline{\text{TEST2}}$ is active
IrDA Interface		
P_ITX	O	Infrared Data Out: This pin provides the serial transmit data from the IrDA codec to external IR Data Transceiver. This function is activated when the IrDA interface is enabled from PERIPHEN I/O Register.
P_IRX	I	Infrared Data-In: This pin provides the serial receive data input from the external IR Data Transceiver to IrDA codec. This function is activated when the IrDA interface is enabled from PERIPHEN I/O Register.
Other Signals		
GND	V	Ground
VCC	V	3.3V power supply
$\overline{\text{RST}}$	I	Reset. A low on this pin for two machine cycles, while the oscillator is running, resets the device.
XTAL1	I	Oscillator Input. Input to the inverting oscillating amplifier. A 12 MHz clock oscillator should be applied.
XTAL2	O	Oscillator Output. Output of the inverting oscillator amplifier.
LFT	I	Master clock PLL LFT pin
UXTALI	I	UART BRG Oscillator Input. Input to the UART oscillator amplifier.
UXTALO	O	UART BRG Oscillator Output. Output of the UART oscillator amplifier.
LFTU	I	UART clock PLL LFT pin

Note: 1. Any signal with an OVERLINE indicates that it is an active low signal.

Functional Description

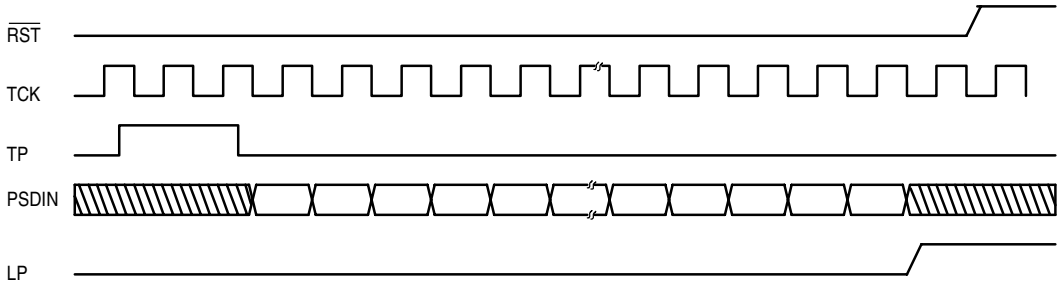
Bootstrap ROM and Program Modes

AT76C711 offers a variety of program modes that allow the user not only to upload the microcode to internal program SRAM but also to upgrade the system firmware that is contained in a serial AT45DB011 (or larger) Flash. AT76C711 supports one slave and three master program modes.

Slave Program Mode

The chip enters the slave program mode while in the reset state ($\overline{RST}$ active low) when it detects a positive edge transition of TP signal. The timing diagram of the procedure is depicted below in Figure 4.

Figure 4. Slave Program Mode Timing Diagram



Master Program Modes

On power-up or after a system reset, the bootstrap code traces the value of the PM0, PM1 signals and executes the respective task according to Figure 4. After the execution of any of the following tasks, the chip enters the normal mode and starts running the code loaded in the internal program SRAM.

PM0	PM1	Task
1	x	SPI program mode: The internal program SRAM is loaded from the external serial Flash through the SPI.
0	0	USB program mode: The host downloads the code to the internal program SRAM using the DFU protocol.
0	1	USB program mode with firmware upgrade: The host downloads pages of code to the internal program SRAM, which are then stored to the external SPI Flash.

USB Hardware Block

USB Function Interface

The USB function interface consists of a Serial Interface Engine (SIE), a Serial Bus Controller (SBC) and a System Interface (SI). The SIE performs the clock/data separation, NRZI encoding and decoding, bit insertion and deletion, CRC generation and checking and the serial-parallel data conversion. The SBC consists of a protocol engine and a USB device with one control endpoint (EP0), four programmable endpoints, each with one 2 x 64-byte dedicated double-buffered FIFO and one programmable endpoint with one 2 x 16-byte double-buffered FIFO. Each EP can be programmed as isochronous, bulk or interrupt and can be configured either as IN or OUT. A pair endpoint address scheme is also supported for the first four programmable endpoints. According to this scheme, two endpoints may have the same address, provided one of them has been configured as IN and the other as OUT. The SBC manages the device address, monitors the status of the transactions, manages the FIFOs and communicates to the microcontroller through a set of status and control registers. The SI connects the SBC to the microcontroller and provides a DMA mechanism for transferring data between the DPRAM and the endpoint buffers.

USB Function Controller

The function controller is implemented in the microcontroller's firmware.

USB Interrupt Handling

All interrupt signals from the USB functions are consolidated into a single interrupt line, which is input to the interrupt controller of the AVR. The following sections describe all the interrupt sources of the USB 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.

Interrupt	Description
Function EP0 Interrupt	See "Control Transfers at Function EP0" for details.
Function EP1 Interrupt	For an OUT endpoint 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	Whenever USB Hardware decodes a valid Start of Frame
EXT RSM	The USB Hardware has received a remote wake-up request.
RCVD RSM	The USB Hardware has received resume signaling. The processor's firmware should take the function out of the suspended state.
SUSP	The USB Hardware 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.

Interrupt Priority

The USB macro interrupt priority is defined below.

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

Endpoint Interrupt

Endpoint interrupts are triggered by setting or clearing one or more bits in the Control and Status registers of an endpoint. These interrupts are caused by events during packet transactions and are different for control and non-control endpoints. The interrupts are described below, with respect to the Control and Status register bit definitions. Please refer to the “Endpoint Control and Status Register” definition on page 55.

Interrupt for Non-control Endpoints

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

Interrupt for Control Endpoints

1. RX OUT Packet set (0 -> 1)
2. RX SETUP set (0 -> 1)
3. TX Packet Ready clear (1 -> 0)
4. TX Complete set (0 -> 1)

Serial Interface Engine

The SIE performs the following functions:

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

Function Interface Unit

The Function Interface Unit (FIU) provides the interface between the processor and the SIE. It manages transactions at the packet level with minimal intervention from the processor and contains the endpoints' buffers.

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 select direction and ping-pong capability. After being 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 will signal 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 zero (0) or more DATA IN or DATA OUT packets for which the processor must either supply or receive the data.

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 FIFO0
		11. Parse data If Set Control Direction Fill FIFO with data Set TX_Packet_Ready If Control Write Phase: Clear Control Direction If No Data Stage Phase: Clear Control Direction Set Data_End bit If Unsupported Command: Set FORCE_STALL bit 12. Clear RX_SETUP bit 13. 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 8. If SET_ADDRESS, write to FADDR
		9. Clear TX_Complete bit

Control Transfers at Function EP0 (Continued)

Legend: DATA1/DATA0 = Data packet with DATA1 or DATA0 PID
DATA 1 (0) = Zero length DATA1 packet

Host	USB Macro	Microcontroller
		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 Set TX_Packet_Ready else Set SET_FORCE_STALL 11. SET UIAR (EP0 INTA)
STATUS/Early STATUS Stage with READ DATA Stage		
1. [SYNC]-[OUT] 2. [SYNC]-[DATA1(0)]	3. If TX_Complete = 0 Send [SYNC]-[ACK] Set RX_OUT else Send [SYNC]-[NACK]	
		4. INTERRUPT
		5. Read UISR 6. Read CSR 7. Clear RX_OUT 8. Set Data_End

Control Transfers at Function EP0 (Continued)

Legend: DATA1/DATA0 = Data packet with DATA1 or DATA0 PID
DATA 1 (0) = Zero length DATA1 packet

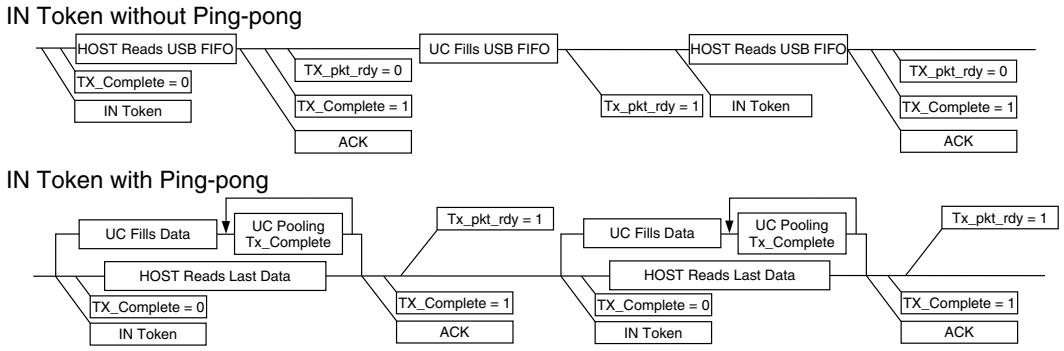
Host	USB Macro	Microcontroller
		9. Set FORCE_STALL **COMMENT:** A SETUP token will clear Data End. Not cleared by firware in case host retries 1 through 3.
		10. SET UIAR (EP0 INTA)
DATA Stage, Control WRITE		
1. [SYNC]-[OUT] 2. [SYNC]-[DATA1/DATA0]	3. Data are put in FIFO 4. If CRC OK, send [SYNC]-[ACK] 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] 3. If CRC OK, Send [SYNC]-[ACK]	2. Send DATA1(0) 4. Set TX_Complete bit	5. INTERRUPT 6. Read UISR 7. Read CSR 8. Clear TX_Complete bit 9. Clear Data_End bit 10. Set FORCE_STALL bit 11. SET UIAR (EP0 INTA)

Interrupt and Bulk IN Transfers

1. The USB Hardware automatically starts the endpoint in receive mode and NAKs all IN tokens as long as bit CSR[TX Packet Ready] is cleared.
2. The processor checks bit CSR[TX Packet Ready]. If it is “0”, writes the data into the FIFO, then sets CSR[TX Packet Ready].
3. 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 bit CSR[TX Packet Ready], signaling a successful completion to the processor.

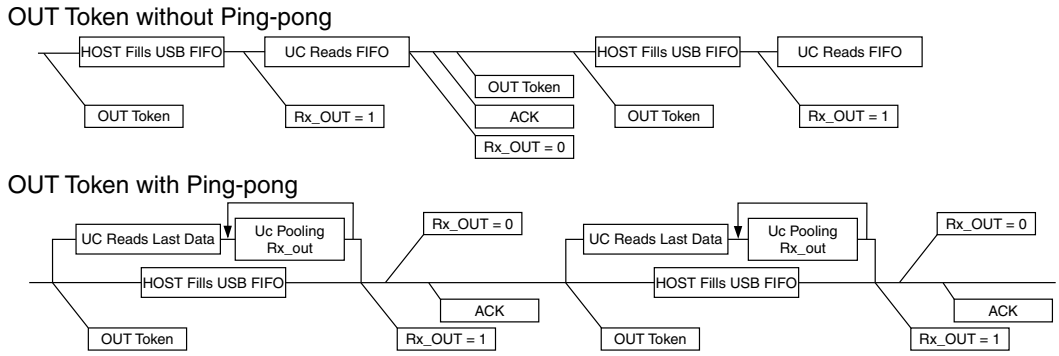
Figure 5. IN Token with and without Ping-pong



Interrupt and Bulk OUT Transfers

The USB Hardware automatically starts the endpoint in receive mode. When an OUT token is received and if 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 CSR[RX OUT Packet] is set, the USB Hardware responds with a NAK to the incoming OUT token: The processor checks 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. OUT Token with and without Ping-pong



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 zero

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).

Suspend

A USB device enters the suspend mode only when requested by the USB host through bus inactivity for at least 3 ms. The USB Hardware 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 Hardware that it can now enter the suspend mode by writing “1” to the “sleep mode” USB_Macro input pin. At this moment, the “Suspend2SIE” output pin is activated and the oscillator, PLL and other peripherals should be disabled.

Resume

Resume is signaled by a J- to K-state transition at the USB port. The USB Hardware 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. Then the firmware clears the RCVD RSM bit.

Remote Wake-up

While the USB peripheral is in suspend mode, resuming is also possible through the remote wake-up feature. Remote wake-up is invoked due to an external event (such as the detection of a key pressing in a keyboard) and is denoted by the “Ext_int” USB_Macro pins (active high). This action, in turn, enables the oscillator and the USB Hardware. The USB Hardware sets the associated flag of UISR and the EXT RSM bit of SPRSR. These generate two interrupts to the processor: Ext_int and RSM_int. The processor starts executing where it left off and services the interrupt. Then the firmware clears the EXT RSM bit and EXT[0-3] bit.

If the remote wake-up feature is enabled and the USB bus remains idle for a period of 5 ms (already 3 ms in the suspend mode), the resume signal is sent to the host during the next 10 ms and the RSMINPR bit of the Global State Register is set.

AVR Microcontroller

The AT76C711 chip is based on the AVR architecture and includes many of the features of the AVR AT90S8515 microcontroller. All peripherals, apart from SPI and timers, are memory mapped to the data address space.

Interrupt Handling

The interrupt vector table of AT76C711 is shown below.

Table 1. AT76C711 Interrupt Vectors

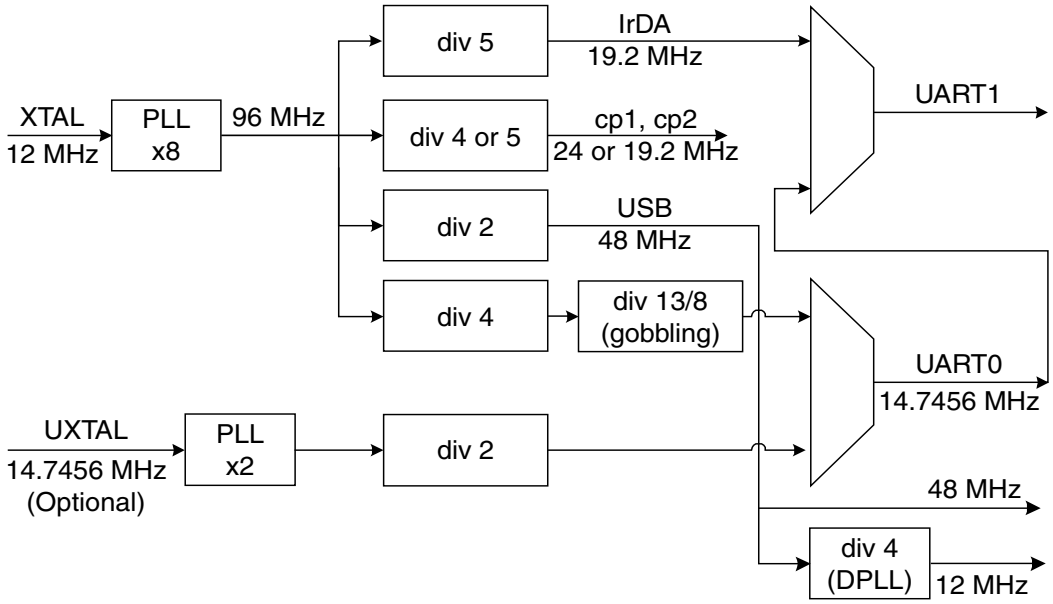
Vector #	Program Address	Source	Interrupt Definition
1	$0000	RESET	Hardware Pin and Watchdog Reset
2	$0001	SUSP/RESM	USB Suspend and Resume
3	$0002	INT0	External Interrupt Request 0
4	$0003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$0004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$0005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$0006	TIMER1 OVF	Timer/Counter1 Overflow
8	$0007	TIMER0 OVF	Timer/Counter0 Overflow
9	$0008	SPI, STC	SPI Serial Transfer Complete
10	$0009	TDMAC	Tx DMA Termination
11	$000A	UART0 INT	UART0 Interrupt Request
12	$000B	RDMAC	Rx DMA Termination
13	$000C	USB Hardware	USB Hardware Interrupt
14	$000D	UART1 INT	UART1 Interrupt Request
15	$000E	INT1	External Interrupt Request 1

Oscillator and Clock Generator

AT76C711 has two on-chip crystal oscillators. The first one is the main oscillator and is used to generate the clocks of the AVR CPU and the 48 MHz clock of the USB core. The nominal value of this oscillator should be 12 MHz. After the initial reset, the default CPU rate is 24 MHz. Dividing the 48 MHz clock appropriately, an internal clock of 14.746 MHz is produced, which can be used for generating standard modem baud rates with a deviation of 1.6 percent.

Alternatively, if a strict baud rate is required, a dedicated oscillator for the UART block is provided. The Clock Tree Circuit is shown in Figure 7. The output pins of the crystal oscillators are not designed to drive any external circuits. Instead of using crystals, either oscillator's input pin can also be driven by an external clock signal.

Figure 7. Clock Tree Circuit



UART0

The main features of UART0 are:

- Programmable Baud Rate Generator
- 16-byte FIFO at the Receiver Side
- Parity, Framing and Overrun Error Detection
- Line Break Generation and Detection
- Automatic Echo, Local Loopback and Remote Loopback Channel Modes
- Interrupt Generation
- Two Dedicated Controller Channels
- 5-, 6-, 7- and 8-bit Character Length
- Maximum Rate 921.6K Baud
- Interface to a DMA Controller for Fast Data Transfers to/from the DPRAM

The input to the baud rate generator is selectable between a 14.746 MHz clock (derived from the internal clock generator) or the dedicated UART oscillator. Both the DMA controller and the AVR processor can have access to the UART registers. The arbitration of the UART memory bus is implemented internally to the DMA controller.

Receiver

The UART detects the start of a received character by sampling the RxD signal until it detects a valid start bit. A low level (space) on RxD 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. Hence a space that is longer than 7/16 of the bit period is detected as a valid start bit. A space that 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 RxD 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. Therefore, 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.

Receive FIFO Operation

The 16-byte receive data FIFO is enabled by the (US_FCR) bit 0. The user can set the receiver trigger level. The receiver FIFO section includes a time-out function to ensure data is delivered to the external CPU. An interrupt is generated whenever the Receive Holding Register (US_RHR) has not been read after the loading of a character or if the trigger level has been reached.

Time-out

This function allows an idle condition on the RxD line to be detected. The maximum delay for which the UART should wait for a new character to arrive while the RxD line is inactive (high level) is programmed in US_RTO (Receiver Time-out). When this register is set to “0”, no time-out is detected. Otherwise, the receiver waits for a first character and then initializes a counter, which decrements at each bit period and is reloaded at each byte reception. When the counter reaches “0”, the time-out bit (bit 6) in US_CSR is set. The user can restart waiting for a first character by setting the start time-out bit (bit 4) of US_CR Register. The time-out duration is:

Duration = Value x _ 4 _ x _ Bit Period

Receive Break

The break condition is detected by the receiver when all data, parity and stop bits are low. At the moment of the low stop bit detection, the receiver asserts receive break (bit 2) in US_CSR. The end of receive break is detected by a high level for at least 2/16 of the bit period. Receive break (bit 2) is also set after the end of break has been detected.

Transmitter

The start bit, data bits, parity bit and stop bits are serially shifted, with the least significant bit first, on the falling edge of the serial clock. The number of data bits is selected in the character length field, (bits 7 and 6) in US_PMR. The parity bit is set according to the parity type bit 1 field in US_PMR. The number of stop bits is selected in the number of stop (bits 4, 5) field in US_MR. When a character is written to US_THR (Transmit Holding), it is transferred to the Shift Register as soon as it is empty. When the transfer occurs, the transmit ready (bit 1) in US_CSR is set until a new character is written to US_THR. If the Transmit Shift Register and US_THR are both empty, the transmitter empty (bit 7) in US_CSR is set.

Time-guard

The time-guard function allows the transmitter to insert an idle state on the TxD line between two characters. The duration of the idle state is programmed in US _TTG (Transmitter Time-guard). When this register is set to “0”, no time-guard is generated. Otherwise, the transmitter holds a high level on TxD after each transmitted byte during the number of bit periods programmed in US_TTG.

Transmit Break

The transmitter can generate a break condition on the TxD line when the Start Break command (bit 0) is set in US_CR (Control Register). In this case, the characters present in US_THR and in the Transmit Shift Register are completed before the line is held low. To remove this break condition on the TxD line, the Stop Break command (bit 1) in US_CR must be set. The UART generates minimum break duration of one character length. The TxD line then returns to high level (idle state) for at least 12 bit periods to ensure that the end of break is correctly detected. Then the transmitter resumes normal operation.

Interrupt Generation

Each status bit in US_CSR has a corresponding bit in US_IER (Interrupt Enable) that controls the generation of interrupts by asserting the UART interrupt line. Any of the Pacity, Framing or overrun Error condition generate a line? Error Interrupt. Interrupt sources are given in the register description section.

Channel Modes

The UART can be programmed to operate in three different test modes using the field Channel Mode (bits 6 and 7) in US_MR. Automatic echo mode allows bit-by-bit retransmission. When a bit is received on the RxD line, it is sent to the TxD line. Programming the transmitter has no effect.

The local loopback mode allows the transmitted characters to be received. TxD and RxD pins are not used and the output of the transmitter is internally connected to the input of the receiver. The RxD pin level has no effect and the TxD pin is held high, as in the idle state.

The remote loopback mode directly connects the RxD pin to the TxD pin. The transmitter and the receiver are disabled and have no effect. This mode allows bit-by-bit retransmission.

UART1 – IrDA Codec

The IrDA codec provides an IrDA 1.0 standard interface. It is connected to the SIN and SOUT of UART1. The IRDAMOD Register of the I/O register set selects the operation mode of the IrDA codec.

The EN bit activates the module. When it is deactivated, the UART signals are connected directly to the external interface, bypassing the IrDA codec block. Otherwise, it encodes the outgoing and decodes the incoming data from and to the UART1 respectively, according to the IrDA 1.0 standard (3/16 modulation).

The MODE bit gives the user the capability to change the modulation scheme from 3/16 pulse width to 4/16.

The UART settings should be half-duplex to avoid signal interference. The UART1 register set is similar to that of UART0: its base address is 2030/hex.

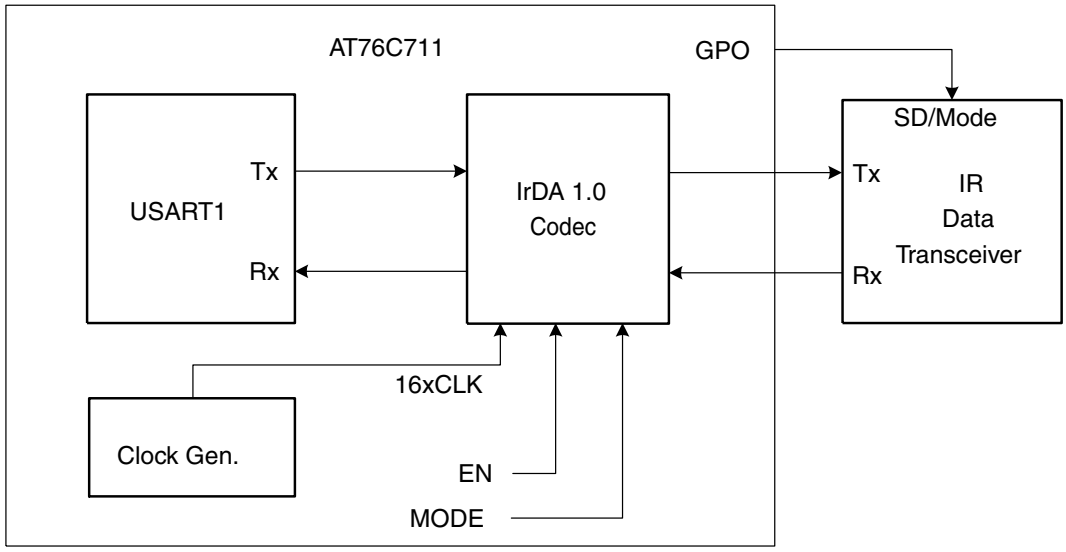
The IrDA codec transmits a 3/16 pulse width on zeros (0) and nothing on ones (1). In receive operations, it extends the incoming zeros to 16/16 pulse and feeds them reversed to the UART1 SIN (see “IrDA Serial Infrared Physical Layer Specification” at www.irda.org/standards).

An internally generated clock of 19.2 MHz makes it possible to use the IrDA module for up to 1.2 Mbps transfer rates (much higher than the typical 115.2 Kbps). This capability offers a simple solution for high-speed infrared communications.

The input to the baud rate generator of UART1 is selectable between the 19.2 MHz clock and the clock of the dedicated oscillator for standard modem rates.

A typical application of the AT76C711 connection to an external IR Data Transceiver when the IrDA module is enabled is shown in Figure 8. The GPO is a general-purpose output (can be provided from the data ports).

Figure 8. AT76C711 Connection to External IR Data Transceiver, IrDA Module Enabled



DMA Controller

The DMA controller is able, under firmware control, to transfer data between the DPRAM and the UART, without the intervention of the processor. The DMA controller will interrupt the processor as soon as it transfers the processor-preprogrammed number of bytes. During data transfers from the DMA controller, the Transmit and Receive DMA Status registers are updated with possible errors indicated by the UART. These status registers can be read by the processor, after the DMA controller's interrupt at the end of a block transfer, to report if the block transfer was error free or not.

The DMA controller is programmed by the processor to transfer blocks of data between the DPRAM and the UART core. In addition, the UART can be accessed by the processor only through the DMA controller module. Thus, data transfers between the processor and other memory devices are not interrupted when DMA transfers occur.

Segmentation and reassembly of the transmitted/received packets through the UART are also executed with the aid of the DMA controller, under firmware control. Reassembly of network packets is implemented by storing the USB packets from a certain endpoint in successive address spaces. The DMA controller is then programmed to consecutively transfer the bytes of the reassembled packet that has been formed. On the other hand, during packet reception from the UART, the processor can program the DMA controller to read a certain number of bytes from the UART's FIFO and store them at a given target address, depending on the packet header. After the end of the DMA transfer, an interrupt is issued by the DMA controller to signal the processor to check for possible errors during this transfer and forward the packet to the EPs of the USB interface.

The DMA controller is programmed with the characteristics of a DMA transfer before it is enabled. The only information that is required is the DPRAM target address and the packet length, because it is already aware of the segment boundaries in order to perform wraparound inside the corresponding segment. In addition, status registers provide information related to errors encountered during a DMA transfer. Two of the above sets of registers are implemented, one for each direction.

Transmit and receive DMAs are performed by polling the TXRDY and RXRDY signals of the UART. A DMA operation consists of reading the UART status registers and accessing the Data Hold Register. Transmit and receive DMA operations can take place simultaneously.

Whenever a receive DMA operation is terminated, either normally or by a Receive Character Time-out Interrupt from the UART or forced by the firmware, an internal RDMAC interrupt signal from DMA is issued to the processor. After that, the firmware should read RXTPLL and RXTPLM to be informed about the exact number of the received bytes, possible errors during DMA and the reason for the DMA termination.

After a transmit DMA termination, either normally or forced by firmware, an internal TDMAC interrupt signal from the DMA is issued to the processor. After that, the firmware should read TXTPLL and TXTPLM to be informed about the exact number of the transmitted bytes and the reason for DMA termination.

DPRAM Organization

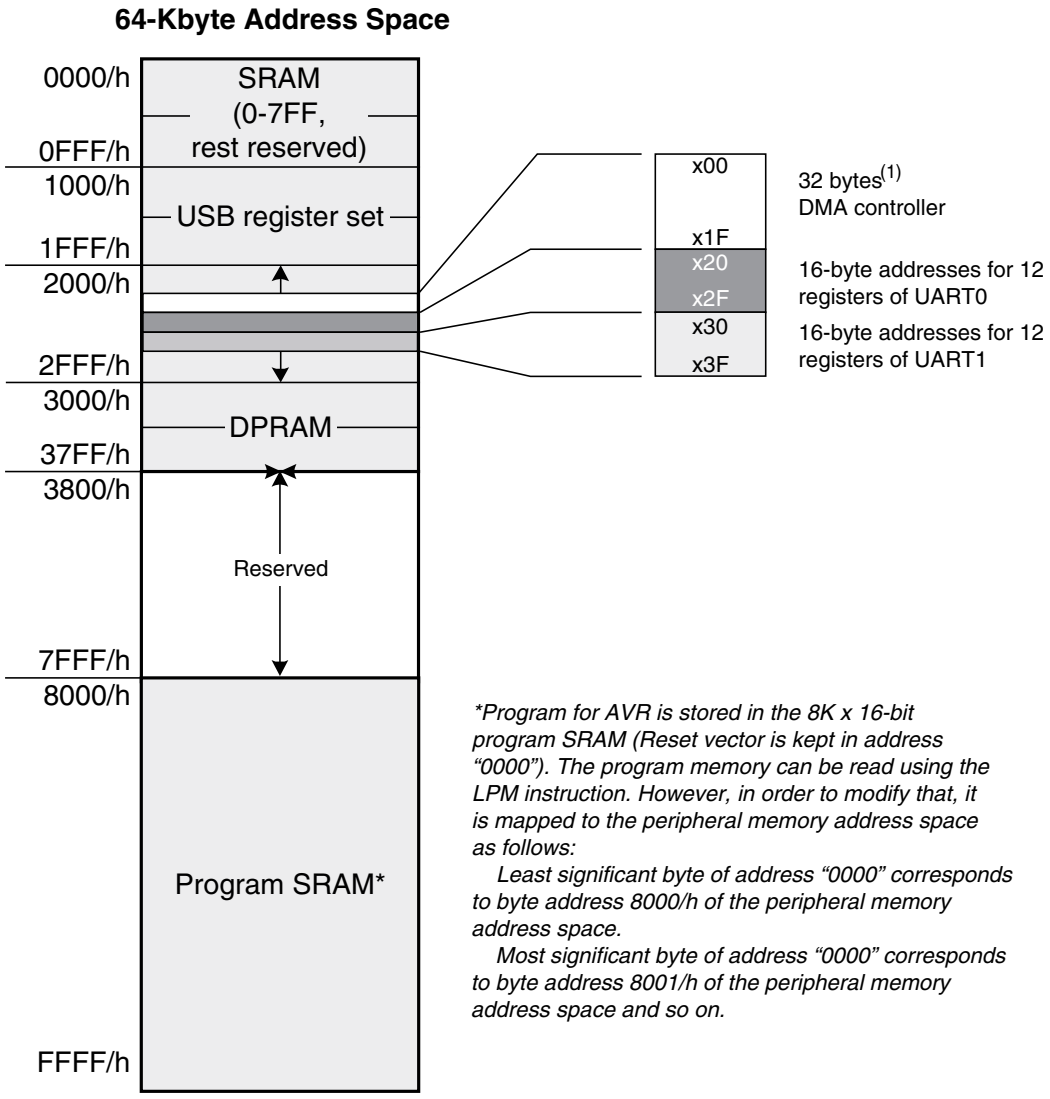
DPRAM organization is related to the length of the packets transferred through UART. The programmer can define segments in the DPRAM address space from DPORG Register of the DMA controller. Memory segmentation facilitates wraparound during DMA transfers. According to the number of segments, the DMA controller can determine the END and START address of each segment so it doesn't need to be informed about the segment boundaries each time a transfer is enabled. The default state is the DPRAM being unsegmented. Details of the possible DPRAM segmentation schemes are given at the description of the DPORG Register of the DMA controller.

Mapping Allocations

The AVR uses a 16-bit address bus to have access to 64-Kbyte memory locations. In AT76C711 design this memory is shared among the various peripherals as shown in Figure 9.

Address Space	Size	Module
0000 - 07FF	2048 bytes	SRAM
0800 - 0FFF	2048 bytes	Reserved
1000 - 1FFF	4096 bytes	USB (not all of the locations are used)
2000 - 201F	32 bytes	DMA controller
2020 - 202F	16 bytes	UART0
2030 - 203F	16 bytes	UART1
2040	1 byte	Program Memory Control bit
2041 - 2FFF	4031 bytes	Reserved
3000 - 37FF	2048 bytes	DPRAM
3800 - 7FFF	18432 bytes	Reserved
8000 - 7FFF	16384 bytes	Program SRAM
7FFF - FFFF	16384 bytes	Reserved

Figure 9. Memory Allocation for On-chip Resources



Note: 1. These register blocks are mapped to any 256-byte boundary (x00) in address space 2000 - 2FFF/h.

I/O Memory

The I/O space definition of AT76C711 is shown in Table 2. 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 will be followed in the rest of this document. A more detailed description of the I/O memory space is given in the sections that follow.

Table 2. AT76C711 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
$39($59)	EIMSK	External Interrupt Mask Register
$37($57)	TIMSK	Timer Interrupt Mask Register
$36($56)	TIFR	Timer Interrupt Flag Register
$35($55)	MCUCR	MCU General Control Register
$34($54)	MCUSR	MCU Status Register
$33($53)	TCCR0	Timer0 Control Register
$32($52)	TCNT0	Timer0 (8 bits)
$31($51)	PRELD	Pre-load Register
$2F($4F)	TCCR1A	Timer1 Control Register A
$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 Register Low Byte
$21($41)	WDTCR	Watchdog Timer Control Register
$20($40)	IRDAMOD	IrDA Control Register
$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)	CLK_CNTR	Clock Control Register

Table 2. AT76C711 I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$13($33)	PERIPHEN	Peripheral Enable Register
$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

The AVR Status Register – SREG

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

- **Bit 7 – I: Global Interrupt Enable**

When set, the interrupts are enabled. The individual interrupt enable control is performed in the individual mask registers. This bit is cleared by USB Hardware after an interrupt has occurred and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 – T: Bit Copy Storage**

Bit load (BLD) and bit store (BST) instructions use the T-bit as source and destination for the operated bit.

- **Bit 5 – H: Half-carry Flag**

Indicates a half-carry in some arithmetic operations.

- **Bit 4 – S: Sign Bit, S = N XOR V**

Is an exclusive OR between the negative flag N and the two's complement overflow flag V.

- **Bit 3 – V: Two's Complement Overflow Flag**

Supports two's complement arithmetic.

- **Bit 2 – N: Negative Flag**

When set, indicates a negative result in arithmetic and logic operations.

- **Bit 1 – Z: Zero Flag**

When set, indicates a zero result after the different arithmetic and logic operations.

- **Bit 0 – C: Carry Flag**

When set, indicates a carry in the arithmetic or logic operations.

The Stack Pointer – SP

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	

The MCU Control Register – MCUCR

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	–	**SE**	**SM1**	**SM0**	–	–	–	**MCUCR**
Read/Write	R	R	R/W	R/W	R/W	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The MCU Control Register is a R/W 3-bit register with zero initial value. It consists of the following bits:

- **Bits 7, 6 – Reserved Bits**

Always read as zero.

- **Bit 5 – SE: Sleep Enable**

When set, enables the MCU to enter sleep mode when the SLEEP instruction is executed.

- **Bits 4, 3 – SM1/SM0: Sleep Mode Select Bits**

These bits select between the three available sleep modes as shown in the following table.

- **Bits 2, 1, 0 – Reserved Bits**

These bits always read as zero

Table 3. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle Mode
0	1	Reserved
1	0	Power-down
1	1	Power Save

MCU Status Register – MCUSR

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	–	–	–	–	–	–	**RESWD**	**PORF**	**MCUSR**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	See bit description		

The MCUSR is a 2-bit R/W register that provides information on which reset source caused an MCU reset.

- **Bits 7..2 – Reserved Bits**

Always read as zero.

- **Bit 1 – REWD**

This flag indicates that an external or power-on reset has occurred.

- **Bit 0 – PORF: Power-on Reset Flag**

The following table shows the value of these two bits after the three modes of reset.

Table 4. 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 reset can be found by using the following truth table.

Table 5. Reset Source Identification

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

The External Interrupt Mask Register – EIMSK

Bit	7	6	5	4	3	2	1	0	
$39 ($59)	–	–	–	–	POL1	POL0	INT1	INT0	EIMSK
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This is a 4-bit R/W register. Its initial value is zero. It is used for masking the external interrupts.

- **Bits 7..4 – Reserved Bits**

Always read as zero.

- **Bit 3 – POL1**

Polarity of external interrupt 1. INT1 is active high when this bit is low.

- **Bit 2 – POL0**

Polarity of external interrupt 0. INT0 is active high when this bit is low.

- **Bit 1 – INT1**

If it is set and the 1 bit in the Status Register is set, the external pin interrupt 1 is enabled.

- **Bit 0 – INT0**

If it is set and the 1 bit in the Status Register is set, the external pin interrupt 0 is enabled.

The Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$37 ($57)	TOIE1	OCIE1A	OCIE1B	–	TICIE1	–	TOIE0	–	TIMSK
R/W	R/W	R/W	R/W	R	R/W	R	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

External interrupts should be acknowledged using general-purpose output pins.

This is an 8-bit R/W register with zero initial value, used for masking the internal timer interrupts.

- **Bit 7 – TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When this bit is set and the I-bit in the Status Register is one, 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.

- **Bit 6 – OCIE1A: Timer/Counter1 Output Compare A Match Interrupt Enable**

When this bit is set and the I-bit in the Status Register is one, 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/Counter1 is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 5 – OCIE1B: Timer/Counter1 Output Compare B Match Interrupt Enable**

When this bit is set and the I-bit in the Status Register is one, 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/Counter1 is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 4 – Reserved Bit**
- **Bit 3 – TICIE1: Timer/Counter1 Input Capture Interrupt Enable**

When this bit is set and the I-bit in the Status Register is one, the Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector $0016) is executed if a capture event occurs on pin PD2. The Input Capture flag in Timer/Counter1 is set in the Timer/Counter Interrupt Flag Register – TIFR.

- **Bit 2 – Reserved Bit**
- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When this bit is set and the I-bit in the Status Register is one, 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/Counter1 Interrupt Flag Register – TIFR.

The Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$36 ($56)	TOV1	OCFA	OCFB	–	ICF1	–	TOV0	–	TIFR
R	R/W	R/W	R/W	R	R/W	R	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 6 – OCFA: Output Compare Flag A**

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”.

- **Bit 5 – OCFB: Output Compare Flag 1B**

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”.

- **Bit 4 – Reserved Bit**
- **Bit 3 – ICF1: Input Capture Flag**

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”.

- **Bit 2 – Reserved Bit**
- **Bit 1 – TOV0: Timer/Counter1 Overflow Flag**

The TOV0 is set when an overflow occurs in Timer/Counter0. This flag is cleared when written with a logic “1”.

- **Bit 0 – Reserved Bit**

The Timer/Counter0 Control Register – TCCR0

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	–	–	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Pre-load Register – PRELD

Bit	7	6	5	4	3	2	1	0	
$31 ($51)	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

An 8-bit R/W register with zero initial value. The contents of this register are loaded to Timer/Counter0 (TCNT0) after an overflow of Timer/Counter0 occurs.

The Timer/Counter1 Control Register A – TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F ($4F)	COM11	COM1A0	COM1B1	COM1B0	–	–	–	–	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This is an 8-bit R/W register with zero initial value. It controls the action taken by the specific output pin on compare match A or B that supports Timer1. The functionality of the bits is as follows:

- **Bits 7, 6 – COM1A1, COM1A0: Compare Output Mode A**

These bits cause a specific action for the output compare pin PD6, as shown in Table 6, for Timer0.

- **Bits 5, 4 – COM1B1, COM1B0**

The same holds for Compare Output Mode B and output compare pin PD7, as in the previous case for the Compare Output Mode A.

- **Bits 3..0 – Reserved Bits**

Table 6. Compare Mode Select[1]

COM1X1	COM1X0	Description
0	0	Timer disconnected from output pin
0	1	Toggle output pin
1	0	Clear output pin
1	1	Set output pin

Note: 1. X = A or B

The Timer/Counter Register B – TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E ($4E)	ICNC1	ICES1	–	–	CTC1	CS12	CS11	CS10	TCCR1A
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This is an 8-bit R/W register with zero initial value.

- **Bit 7 – ICNC1: Input Capture1 Noise Canceler (4 CKs)**

When this bit is zero, the input canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the input capture pin PB2, 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.

- **Bit 6 – ICES1: Input Capture1 Edge Select**

When this bit is cleared, the Timer/Counter1 contents are transferred to the Input Capture Register – ICR1 on the falling edge of the input capture pin, PB2. When it is “1”, the contents are transferred on the rising edge.

- **Bits 5, 4 – Reserved Bits**

Always read as zero.

- **Bit 3 – CTC1: Clear Timer/Counter1 on Compare Match**

When it is “1”, the Timer/Counter1 is reset to $0000 after a Compare A match. If it is cleared, the Timer/Counter1 continues counting after a Compare A match.

- **Bits 2..0 – CS12, CS11 and CS10: Clock Select1, Bits 2, 1 and 0**

These bits select prescaling source for the Timer/Counter1 according to the following table.

Table 7. Timer1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Timer1 is stopped
0	0	1	TCK1
0	1	0	TCK1/8
0	1	1	TCK1/64
1	0	0	TCK1/256
1	0	1	TCK1/1024
1	1	0	External Pin PB1, rising edge
1	1	1	External Pin PB1, falling edge

The Timer/Counter1 – TCNT1H and TCNT1L

Bit	15	14	13	12	11	10	9	8	
$2D ($4D)	**MSB**								**TCNT1H**
$2C ($4C)								**LSB**	**TCNT1L**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This Timer/Counter consists of 16 bits and is made by two 8-bit R/W registers with initial value of zero, namely TCNT1H and TCNT1L. 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 interrupt routines perform accesses to registers using TEMP, interrupts must be disabled during access from the main program.

- **TCNT1 Timer/Counter1 Write**

When the CPU writes to the high byte (TCNT1H), the written data are 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 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.

The Timer/Counter1 Output Compare Register B – OCR1BH and OCR1BL

Bit	15	14	13	12	11	10	9	8	
$29 ($49)	**MSB**								**OCR1BH**
$28 ($48)								**LSB**	**OCR1BL**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

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

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

The Timer/Counter1 Input Capture Register – ICR1H and ICR1L

Bit	15	14	13	12	11	10	9	8	
$27 ($47)	**MSB**								**ICR1H**
$26 ($46)								**LSB**	**ICR1L**
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

This Timer/Counter Output Compare Register consists of 16 bits and is made by two 8-bit R/W registers with initial value of zero, namely ICR1H and ICR1L.

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

The Watchdog Timer Control Register – WDTCR

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	**–**	**–**	**–**	**WDTOE**	**WDE**	**WDP2**	**WDP1**	**WDP0**	**WDTCR**
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This is an 8-bit R/W register with zero initial value.

- **Bits 7..5 – Reserved Bits**

Always read as zero.

- **Bit 4 – WDTOE: Watchdog Turn Off Enable**

This bit must be set when the WDE bit is cleared. Otherwise, the watchdog will not be disabled. Once set, USB Hardware will clear this bit to zero after four clock cycles.

- **Bit 3 – WDE: Watchdog Enable**

When this bit is set, the watchdog timer is enabled; when it is zero, the watchdog timer is disabled. WDE can only be cleared when the WDTOE is set. To disable an enabled watchdog timer, the following procedure must be followed:

1. In the same operation, write a logic “1” to WDTOE and WDE. A logic “1” must be written to WDE even though it is set to one before the disable operation starts.
2. Within the next four clock cycles, write a logic “0” to WDE. This disables the watchdog.

- **Bits 2..0 – WDP2, WDP1, WDP0: Watchdog Timer Prescaler 2, 1 and 0**

These bits determine the watchdog timer prescaling when the watchdog timer is enabled according to the following table.

Table 8. Watchdog Timer Prescale Register Select

WDP2	WDP1	WDP0	Time-out Period (Cycles)
0	0	0	16K
0	0	1	32K
0	1	0	64K
0	1	1	128K
1	0	0	256K
1	0	1	512K
1	1	0	1024K
1	1	1	2084K

The IRDAMOD Register

Bit	7	6	5	4	3	2	1	0	
$20 ($40)						POL	MODE	EN	PORTA
Read/Write						R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7..3 – Reserved Bits**

Always read as zero.

- **Bit 2 – POL**

Inverts the polarity of P_IRX data input to IrDA encoder.

- **Bit 1 – MODE**

When this bit is set, serial data to and from IrDA codec are modulated according to a 4/16 pulse scheme. Otherwise, a logic “0” is denoted by a positive pulse, which remains high for 3 pulses of the 16x baud clock while a logic “1” is denoted by a low-level signal for 16 pulses of the 16x baud clock.

- **Bit 1 – EN**

When reset, IrDA codec is bypassed.

I/O Ports

Port A

Port A is an 8-bit bi-directional I/O port.

The Port A Data Register – PORTA

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	PORTA7	PORTA6	PORTA5	PORTA4	PORTA3	PORTA2	PORTA1	PORTA0	PORTA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Port A Data Direction Register – DDRA

Bit	7	6	5	4	3	2	1	0	
$1A ($3A)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	DDRA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The bits in the Data Direction Register control the direction of the corresponding pin in Port A.

When bit DDRAX is set, then PAX pin is output; when DDRAX is cleared, PAX is input; X = 0..7.

The Port A Input Pins Address – PINA

Bit	7	6	5	4	3	2	1	0	
$19 ($39)	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	PINA
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	

The Port A Input Pins Address (PINA) is not a physical register. Instead, this address enables access to the physical voltage value on each port pin. It is a read-only address. The pins of the port have high-Z initial value.

Port B

Port B is an 8-bit bi-directional I/O port with internal pull-ups.

The Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
$18 ($38)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This 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 Port B.

When bit DDRBX is set, then PBX pin is input; when DDRBX is cleared, PBX is output; X = 0..7.

The Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
$16 ($36)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	High-Z	

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 pins of the port have high-Z initial value.

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 Timer/Counters are implemented through Port B, according to the following table.

Table 9. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	External Clock Pin for Timer/Counter0
PB1	External Clock Pin for Timer/Counter1
PB2	Input Capture Pin for Timer/Counter1
PB4	SS (SPI Slave Select Input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

Port C

Port C is a 4-bit output port.

The Port C Data Register – PORTC

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	–	–	–	–	PORTC3	PORTC2	PORTC1	PORTC0	PORTC
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Port D

Port D is an 8-bit bi-directional I/O port with internal pull-ups.

The Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
$12 ($32)	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Data Direction Register controls the direction of the corresponding pin in Port D.

When bit DDRDX is set, then PDX pin is input; when DDRDX is cleared, PDX is output; X = 0..7.

The Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
$11 ($31)	DDD7	DDD6	DDD5	DDDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	1	1	1	1	1	1	1	

An 8-bit register with zero initial value, 0xff.

Table 10. Port D Pins Alternate Functions

Port Pin	Alternate Functions
PD0	UART Receive Input
PD1	UART Transmit Output
PD6	External Interrupt Input0 or Output Compare A Pin
PD7	External Interrupt Input1 or Output Compare B Pin

Clock Control Register – CLK_CNTR

Bit	7	6	5	4	3	2	1	0	
$14 ($34)				UOSC	UCK	IrCK			CLK_CNTR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This is a 5-bit R/W register that is used for controlling the clocks of the peripheral components and the speed of the MCU.

- **Bits 7..5 – Reserved**

Always read as zero.

- **Bit 4 – UOSC**

Enables UART oscillator when set.

- **Bit 3 – UCK**

Select clock source for UART module. When cleared, the UART 16xbaud_rate clock is 14.76 MHz; when set and UOSC is also set, the UART oscillator provides the 16xbaud_rate UART clock.

- **Bit 2 – IrCK**

Select clock source for UART - IrDA module. When cleared, the 16xbaud_rate clock of this module is 14.76 MHz; when set and UOSC is also set, the UART oscillator provides the 16xbaud_rate clock for the UART - IrDA module.

- **Bits 1, 0 – Reserved**

These bits should be set to zero

Peripheral Enable Control Register – PERIPHEN

Bit	7	6	5	4	3	2	1	0	
$13 ($33)						IRDA	UART	USB	PERIPHEN
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

A 6-bit R/W register that enables the peripheral components of the system, such as SPI, UART and USB.

- **Bits 7, 6 – Reserved Bits**

Always read as zero.

- **Bits 5..3 – Reserved Bits**
- **Bit 2 – IRDA**

When set, enables the function of IrDA.

- **Bit 1 – UART**

When set, enables the function of UART.

- **Bit 0 – USB**

When set, enables the function of USB (clock enable).

The SPI Control Register – SPCR

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	**SPIE**	**SPE**	**DORD**	**MSTR**	**CPOL**	**CPHA**	**SPR1**	**SPR0**	**SPCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

An 8-bit R/W register with zero initial value.

- **Bit 7 – SPIE: SPI Interrupt Enable**

This bit causes setting of the SPIF bit in the SPSR Register to execute the SPI Interrupt provided that global interrupts are enabled.

- **Bit 6 – SPE: SPI Enable**

When it is set, the SPI is enabled and SS, MOSI, MISO and SCK are connected to pins PB4, PB5, PB6 and PB7.

- **Bit 5 – DORD: Data Order**

When it is “1”, the LSB of the data word is transmitted first. When it is cleared, the MSB of the data word is transmitted first.

- **Bit 4 – MSTR: Master/Slave Select**

This bit selects Master SPI when set and Slave SPI mode when cleared. If SS is configured as input and is driven low while MSTR is set, MSTR will be cleared and SPIF in SPSR will become set. The user will then have to set MSTR to re-enable SPI master mode.

- **Bit 3 – CPOL: Clock Polarity**

When this bit is set, SCK is high when idle. When CPOL is cleared, SCK is low when idle.

- **Bit 2 – CPHA: Clock Phase**

When set, the data is valid in the falling edge of SCK if CPOL = 0, or in the rising edge of SCK when CPOL = 1. When cleared, data are valid in the rising edge of SCK if CPOL = 0 and in the falling edge of SCK if CPOL = 1.

- **Bits 1, 0 – SPR1, SPR0: SPI Clock Rate Select 1 and 0**

These two bits control the SCK rate of the device configured as a master. SPR1 and SPR0 have no effect on the slave. The relationship between the slave and the oscillator clock frequency (f_{cl}) is shown in the following table.

Table 11. Relationship between SCK and the Oscillator Frequency

SPR1	SPR0	SCG Frequency
0	0	f_{cl}/4
0	1	f_{cl}/16
1	0	f_{cl}/64
1	1	f_{cl}/128

The SPI Status Register – SPSR

Bit	7	6	5	4	3	2	1	0	
$0E ($2E)	**SPIF**	**WCOL**	–	–	–	–	–	–	**SPSR**
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This is an 8-bit read register with zero initial value.

- **Bit 7 – SPIF: SPI Interrupt Flag**

When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE in SPCR is set and global interrupts are enabled. Alternatively, the SPIF bit is cleared by first reading the SPI Status Register with SPIF set, then by accessing the SPI Data Register.

- **Bit 6 – WCOL: Write Collision Flag**

The WCOL bit is set if the SPI Data Register (SPDR) is written during a data transfer.

During data transfer, the result of reading the SPDR may be incorrect, and writing to it will have no effect. The WCOL bit (and the SPIF bit) are cleared by first reading the SPI Status Register with WCOL set, and then by accessing the SPI Data Register.

- **Bits 5..0 – Reserved**

Always read as zero.

The SPI Data Register

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	**MSB**							**LSB**	**SPDR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

An 8-bit R/W register with zero initial value. 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.

UART Register Set

The base address for UART registers is 2020/hex. In the following table the register file and its fields are briefly presented. A more detailed description is provided in the following sections.

Table 12. UART Register File and Register Fields

Offset Addr. A[3:0]	Register [default]	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit0
0000	US_RHR	X	X	X	X	X	X	X	X
0000	US_THR	X	X	X	X	X	X	X	X
0001	US_IER	Transmitter Empty Interrupt	Receive Time-out Interrupt			Line Error Interrupt	Receive Break Interrupt	Transmit Holding Register	Receive Holding Register
0010	US_FCR	RCVR Trigger (MSB)	RCVR Trigger (LSB)	0	0	DMA Mode Select	0	RCVR FIFO Reset	FIFO Enable
0011	US_PMR	Character Length (MSB)	Character Length (MSB)	Number of Stop Bits (MSB)	Number of Stop Bits (LSB)	Parity Mode (PTM2)	Parity Mode (PTM1)	Parity Type (PT)	0
0100	US_MR	Channel Mode (CHM1)	Channel Mode (CHM0)	0	0	0	0	0	0
0101	US_CSR	Transmitter Empty	Receive Time-out	Parity Error	Framing Error	Overrun Error	Receive Break	Transmit Holding Register Ready	Receive Holding Register Ready
0110	US_CR	Rx Enable	Reset Status Bit	Tx Enable	Restart Time-out	Tx Reset	Rx Reset	Stop Break	Start Break
0111	US_BL	MSB							LSB
1000	US_BM	MSB							LSB
1001	US_RTO	MSB							LSB
1010	US_TTG	MSB							LSB

Receive Holding Register – US_RHR

Bit	7	6	5	4	3	2	1	0	
$0	MSB							LSB	US_RHR
Read/Write	W	W	W	W	W	W	W	W	
Initial Value	0	0	0	0	0	0	0	0	

Transmit Holding Register – US_THR

Bit	7	6	5	4	3	2	1	0	
$0	MSB							LSB	US_THR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Interrupt Enable Register – US_IER

Bit	7	6	5	4	3	2	1	0	
$1	TXEI	RXTOI	Res	Res	RLEI	RBRI	THRI	RHRI	US_IER
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Transmitter Empty Interrupt**

When set, the interrupt is enabled. When both the Transmit Holding Register (US_THR) and the Transmit Shift Register are empty and the I-bit in the Status Register (SREG) of the MCU is set, an interrupt will occur.

- **Bit 6 – Receive Time-out Interrupt**

When set, the Time-out Interrupt is enabled. When the time-out period for the receiver has passed and the I-bit in the Status Register (SREG) of the MCU is set, an interrupt will occur.

- **Bit 5 – Reserved**
- **Bit 4 – Reserved**
- **Bit 3 – Line Error Interrupt**

This bit, when set, enables the Line Error Interrupt. If the I-bit in the Status Register (SREG) of the MCU is set, an interrupt will occur at a line error.

- **Bit 2 – Receive Break Interrupt**

When set, enables Receive Break Interrupt. If a receive break condition is detected and both this and the I-bit of SREG of the MCU is set, an interrupt will occur.

- **Bit 1 – Transmit Holding Register Interrupt**

When set, indicates that the Transmit Ready Interrupt is enabled. When the contents of the Transmit Holding Register are transferred to the Transmit Shift Register and both this and the I-bit of SREG of the MCU are set, an interrupt occurs.

- **Bit 0 – Receive Holding Register Interrupt**

When set, indicates that the Receive Holding Register Interrupt is enabled. If the data loaded in the Receive Holding Register (US_RHR) are not read or the trigger level has been reached, an interrupt occurs if this bit and the I-bit of the SREG of the MCU are set.

FIFO Control Register – US_FCR

Bit	7	6	5	4	3	2	1	0	
$2	RCVR1	RCVR0	Res	Res	RDMA	Res	FRS	FEN	US_FCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7, 6 – 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 in the following table.

Bit 7	Bit 6	Trigger Level
0	0	1
0	1	4
1	0	8
1	1	14

- **Bit 5 – Reserved**
- **Bit 4 – Reserved**
- **Bit 3 – DMA Mode Select**

When set, the DMA is in burst mode according to the value in US_FCR. When it is cleared, the characters are read one byte each time.

- **Bit 2 – Reserved**
- **Bit 1 – FIFO Reset**

When set, resets the receive FIFO.

- **Bit 0 – FIFO Enable**

When set, enables the 16-byte receive FIFO.

Protocol Mode Register – US_PMR

Bit	7	6	5	4	3	2	1	0	
$3	**CHL1**	**CHL0**	**SBN1**	**SBN0**	**PM1**	**PM0**	**Res**	**LSB**	**US_PMR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7, 6 – Character Length**

These bits determine the character length, according to the following table.

Bit 7	Bit 6	Character Length
0	0	5
0	1	6
1	0	7

- **Bits 5, 4 – Number of Stop Bits**

These bits determine the number of stop bits, according to the following table.

Bit 5	Bit 4	Number of Stop Bits
0	0	1
0	1	1.5
1	0	2
1	1	Reserved

- **Bits 3, 2 – Parity Mode**

These bits determine the parity mode, according to the following table.

Bit 3	Bit 2	Parity Mode
0	0	Normal Parity
0	1	Parity Forced
1	0	No Parity
1	1	Multidrop

- **Bit 1 – Parity Type**

In normal parity mode this bit is used for the determination of parity. In force parity mode this bit is forced to be the parity bit.

- **Bit 0 – Reserved**

Mode Register – US_MR

Bit	7	6	5	4	3	2	1	0	
$4	**CHM1**	**CHM0**	**Res**	**Res**	**Res**	**Res**	**Res**	**Res**	**US_MR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7, 6 – Channel Mode**

These bits determine the channel mode, according to the following table.

Bit 7	Bit 6	Parity Mode
0	0	Normal Mode
0	1	Automatic Echo Mode
1	0	Local Loopback Mode
1	1	Remote Loopback Mode

- **Bit 5 – Reserved**

Control Status Register- US_CR

Bit	7	6	5	4	3	2	1	0	
$5	**TXE**	**RXTO**	**PE**	**FE**	**OE**	**RBR**	**THR**	**RHR**	**US_CSR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Transmitter Empty**

When set, indicates that both the Transmit Holding Register (US_THR) and Transmit Shift Register are empty.

- **Bit 6 – Receive Time-out**

When set, indicates that a receive time-out condition has occurred.

- **Bit 5 – Parity Error**

When set, indicates that a parity error has occurred.

- **Bit 4 – Framing Error**

When set, indicates that a framing error has occurred (start or stop bits have been received with errors).

- **Bit 3 – Overrun Error**

When set, indicates that an overrun error has occurred. This means that the Receive Holding Register is being written with a new value, while the previous one has not been read.

- **Bit 2 – Receive Break**

When set, indicates that a break condition has occurred during reception.

- **Bit 1 – Transmit Holding Register Ready**

When set, indicates that the contents of the Transmit Holding Register have been transferred to the Transmit Shift Register.

- **Bit 0 – Receive Holding Register Ready**

When set, indicates that the Receive Holding Register is full.

Control Register – US_CR

Bit	7	6	5	4	3	2	1	0	
$6	**RXEN**	**RLES**	**TXEN**	**RSTO**	**TXRS**	**RXRS**	**SPB**	**STB**	**US_CR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Rx Enable**

When set, enables the receiver block of UART.

- **Bit 6 – Reset Line Error Status Bit**

When set, resets the PE, FE, OE bits of US_CSR.

- **Bit 5 – Tx Enable**

When set, enables the transmitter block of UART.

- **Bit 4 – Restart Time-out**

When set, resets the time-out counter for a new time-out period.

- **Bit 3 – Tx Reset**

When set, resets the transmit logic.

- **Bit 2 – Rx Reset**

When set, resets the receive logic.

- **Bit 1 – Stop Break**

Break command to the transmit logic. When set, stops break condition.

- **Bit 0 – Start Break**

Break command to the transmit logic. When set, starts break condition.

Baud Rate Register, Low Byte – US_BL

Bit	7	6	5	4	3	2	1	0	
$7	MSB							LSB	US_BL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Baud Rate Register, High Byte – US_BM

Bit	7	6	5	4	3	2	1	0	
$8	MSB							LSB	US_BM
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Receiver Time-out Register – US_RTO

Bit	7	6	5	4	3	2	1	0	
$9	MSB							LSB	US_RTO
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the maximum period for which the UART can wait before a character arrives during the time-out function. This function is disabled when this register is zero. The value of register US_RTO represents bit periods.

Transmitter Time-guard Register – US_TTG

Bit	7	6	5	4	3	2	1	0	
$A	MSB							LSB	US_TTG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The value of this register indicates the delay (in bit periods) that an active transmitter has to interpose between two consecutive character transmissions.

Table 13. Baud Rate Generation Example (Clock = 14,7456 MHz)

Output Baud Rate	User Divisor (16 • clk)		UBM Value (Hex)	UBL Value (Hex)
	(Decimal)	(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
19200	48	30	00	30
38400	24	18	00	18
57.6k	16	10	00	10
76.81	12	0C	00	0C
153.6k	6	06	00	06
307.2k	3	03	00	03
460.8k	2	02	00	02
921.6k	1	01	00	01

DMA Controller Register Set

The base address for the DMA register file is 2000/hex. The following sections describe the bits of the registers of the DMA controller.

TXTADL – Transmit DMA Target Address LSBs

Register Address: 0x2001

Default State: 0x00

Bit	Mnemonic	Description
7	TXDMAEN	Transmit DMA enable
6 - 4	Reserved	
3 - 0	TXTAD[11:8]	Most significant bits of the 12-bit transmit DMA target address

TXPLL – LSBs of Packet Length during Direct Memory Readings

Register Address: 0x2003

Default State: 0x00

Bit	Mnemonic	Description
7 - 0	TXPL[7:0]	Least significant bits of the 11-bit transmit packet length field

TXPLM – MSBs of Packet Length during Direct Memory Readings

Register Address: 0x2004

Default State: 0x00

Bit	Mnemonic	Description
7 - 3	Reserved	
2 - 0	TXPL[10:8]	Most significant bits of the 11-bit transmit packet length field

TXTPLL – LSBs of the Number of Bytes Transmitted during the Last Transmit DMA

Register Address: 0x2005

Default State: 0x00

Bit	Mnemonic	Description
7 - 0	TXTPL[7:0]	Least significant bits of the number of bytes transmitted during the last transmit DMA operation

TXTPLM – MSBs of the Number of Bytes Transmitted during the Last Transmit DMA and Transmit Status Information

Register Address: 0x2006

Default State: 0x00

Bit	Mnemonic	Description
7 - 5		Transmit status information (to be defined)
4	TCOM	DMA has normally terminated after transmitting the requested number of bytes. Cleared when this register is read.
3	TNCOM	Host has ceased a transmit DMA before the requested buffer size is transmitted. Cleared when this register is read.
2 - 0	TXTPL[10:8]	Most significant bits of the number of bytes transmitted during the last transmit DMA operation

RXTADL – Received DMA Target Address LSBs

Register Address: 0x2007

Default State: 0x00

Bit	Mnemonic	Description
7 - 0	RXTAD[7:0]	Least significant bits of the 12-bit receive DMA target address

RXTADMEN – Receive DMA Target Address MSBs, DMA Enable

Register Address: 0x2008

Default State: 0x00

Bit	Mnemonic	Description
7	RXDAMAEN	Receive DMA enable
6 - 4	Reserved	
3 - 0	RXTAD[11:8]	Most significant bits of the 12-bit receive DMA target address

RSPLL – LSBs of Packet Length during Direct Memory Writings

Register Address: 0x2009

Default State: 0x00

Bit	Mnemonic	Description
7 - 0	RXPL[7:0]	Least significant bits of the 11-bit receive packet length field

RXPLM – MSBs of Packet Length during Direct Memory Writings

Register Address: 0x200A

Default State: 0x00

Bit	Mnemonic	Description
7 - 3	Reserved	
2 - 0	RXPL[10:8]	Most significant bits of the 11-bit receive packet length field

RXTPLL – LSBs of the Number of Bytes Received during the Last Receive DMA

Register Address: 0x200B

Default State: 0x00

Bit	Mnemonic	Description
7 - 0	RXTPL[7:0]	Least significant bits of the number of the received bytes during the last receive DMA operation

RXTPLM – MSBs of the Number of Bytes Received during the Last Receive DMA and Receive Status Information

Register Address: 0x200C

Default State: 0x00

Bit	Mnemonic	Description
7	RER	At least one error (parity, framing, overrun) occurred during packet reception. Cleared when a new receive DMA is programmed.
6	Reserved	
5	Reserved	
4	RCOM	DMA has normally terminated after receiving the requested number of bytes. Cleared when this register is read.
3	RNCOM	The processor has ceased DMA before receiving the expected number of bytes. Cleared when this register is read.
2 - 0	RXTPL[10:8]	Most significant bits of the number of the received bytes during the last receive DMA operation

INTCST – Interrupt Control and Status Register

Register Address: 0x200D

Default State: 0x00

Bit	Mnemonic	Description
7 - 4	Reserved	
3	DTXIRQ	This bit indicates that a Transmit DMA Interrupt request is pending. This bit is cleared and the corresponding interrupt is acknowledged when the RXTPLM Register is read.
2	DRXIRQ	This bit indicates that a Receive DMA Interrupt request is pending. This bit is cleared and the corresponding interrupt is acknowledged when the RXTPLM Register is read.
1	TXINTE	Enables Transmit DMA interrupts
0	RXINTE	Enables Receive DMA interrupts

USB Register Set

The USB appears to the AVR just like another peripheral. The USB register file is mapped to the SRAM space. The following table summarizes the USB cell-specific registers.

Table 14. USB Register Set

Register	Address	Default	Function
FRM_NUM_H	0x0FD	xxxxx000	Frame Number High Register
FRM_NUM_L	0x0FC	xxxxx000	Frame Number Low Register
GLB_STATE	0x0FB	xxxxx000	Global State Register
SPRSR	0x0FA	xxxxx000	Suspend/Resume Register
SPRSIE	0x0F9	xxxxx000	Suspend/Resume Interrupt Enable Register
UISR	0x0F7	00000000	USB Interrupt Status Register
UIAR	0x0F5	xxxxx000	USB Interrupt Acknowledge Register
FADDR	0x0F2	00000000	Function Address Register
ENDPPGPG	0x0F1	00000000	Function Endpoint Ping-pong Register
ECR0	0x0EF	0xxx0000	Endpoint0 Control Register
ECR1	0x0EE	0xxx0000	Endpoint1 Control Register
ECR2	0x0EC	0xxx0000	Endpoint2 Control Register
ECR3	0x0EC	0xxx0000	Endpoint3 Control Register
ECR4	0x0EB	0xxx0000	Endpoint4 Control Register
ECR5	0x0EA	0xxx0000	Endpoint5 Control Register
ECR6	0x0E9	0xxx0000	Endpoint6 Control Register
ECR7	0x0E8	0xxx0000	Endpoint7 Control Register
CSR0	0x0DF	x1110000	Endpoint0 Control and Status Register
CSR1	0x0DE	x1110000	Endpoint1 Control and Status Register
CSR2	0x0DD	x1110000	Endpoint2 Control and Status Register
CSR3	0x0DC	x1110000	Endpoint3 Control and Status Register
CSR4	0x0DB	x1110000	Endpoint4 Control and Status Register
CSR5	0x0DA	x1110000	Endpoint5 Control and Status Register
CSR6	0x0D9	x1110000	Endpoint6 Control and Status Register
CSR7	0x0D8	x1110000	Endpoint7 Control and Status Register
FDR0	0x0CF	00000000	FIFO 0 Data Register
FDR1	0x0CE	00000000	FIFO 1 Data Register
FDR2	0x0CD	00000000	FIFO 2 Data Register
FDR3	0x0CC	00000000	FIFO 3 Data Register
FDR4	0x0CB	00000000	FIFO 4 Data Register
FDR5	0x0CA	00000000	FIFO 5 Data Register
FDR6	0x0C9	00000000	FIFO 6 Data Register
FDR7	0x0C8	00000000	FIFO 7 Data Register
FBYTE_CNT0_L	0x0BF	00000000	Byte Count FIFO 0 Register [7:0]

Table 14. USB Register Set (Continued)

Register	Address	Default	Function
FBYTE_CNT1_L	0x0BE	00000000	Byte Count FIFO 1 Register [7:0]
FBYTE_CNT2_L	0x0BD	00000000	Byte Count FIFO 2 Register [7:0]
FBYTE_CNT3_L	0x0BC	00000000	Byte Count FIFO 3 Register [7:0]
FBYTE_CNT4_L	0x0BB	00000000	Byte Count FIFO 4 Register [7:0]
FBYTE_CNT5_L	0x0BA	00000000	Byte Count FIFO 5 Register [7:0]
FBYTE_CNT6_L	0x0B9	00000000	Byte Count FIFO 6 Register [7:0]
FBYTE_CNT7_L	0x0B8	00000000	Byte Count FIFO 7 Register [7:0]
FBYTE_CNT0_H	0x0AF	00000000	Byte Count FIFO 0 Register [10:8]
FBYTE_CNT1_H	0x0AE	00000000	Byte Count FIFO 1 Register [10:8]
FBYTE_CNT2_H	0x0AD	00000000	Byte Count FIFO 2 Register [10:8]
FBYTE_CNT3_H	0x0AC	00000000	Byte Count FIFO 3 Register [10:8]
FBYTE_CNT4_H	0x0AB	00000000	Byte Count FIFO 4 Register [10:8]
FBYTE_CNT5_H	0x0AA	00000000	Byte Count FIFO 5 Register [10:8]
FBYTE_CNT6_H	0x0A9	00000000	Byte Count FIFO 6 Register [10:8]
FBYTE_CNT7_H	0x0A8	00000000	Byte Count FIFO 7 Register [10:8]
SLP_MD_EN	0x100	00000000	Sleep Mode Control
IRQ_EN	0x101	00000000	Master Interrupt Enable
IRQ_STAT	0x102	00000000	Master Interrupt Status
SUSP_WUP	0x103	00000000	Suspend Status
PA_EN	0x104	00000000	Pair Addressing Enable
USB_DMA_ADL	0x105	00000000	DMA Address LO
USB_DMA_ADH	0x106	00000000	DMA Address HI
USB_DMA_PLR	0x107	00000000	DMA Packet Length Requested
USB_DMA_EAD	0x108	00000000	DMA Target Endpoint Address
USB_DMA_PLT	0x109	00000000	DMA Packet Length Transferred
USB_DMA_EN	0x10a	00000000	DMA Enable

FRM_NUM_H: Frame Number High Register

Register Address: 0x0FD

Default State: 0x00

Bit	Field	USB Hardware	Description
7 - 3	Reserved		Reserved and set to 0
2 - 0	FCH[10:8]	W	These are the upper 3 bits of the 11-bit frame number of SOF packet.

FRM_NUM_L: Frame Number Low Register

Register Address: 0x0FC

Default State: 0x00

Bit	Field	USB Hardware	Description
7 - 0	FCL[7:0]	W	These are the lower 8 bits of the 11-bit frame number of SOF packet.

Global State Register

Register Address: 0x0FB

Default State: 0000000000b

Bit	Field	USB Hardware	Description
4 - 7	Reserved		Reserved
3	RSMINPR	W	Set by USB Hardware when a Resume is sent in the USB bus during remote wake-up feature (13 ms).
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_ADDRESS request transaction. The host will use the new address, starting at the next transaction.

SPRSR: Suspend/Resume Register

Register Address: 0x0FA

Default State: xxxxx000b

Bit	Field	USB Hardware	Description
7	Reserved		
6	Reserved		
5	Reserved		
4	Reserved		
3	SOF INT	W	Start of Frame Interrupt
2	EXT RSM	W	Received External Resume. The USB Hardware sets this bit to denote an External Resume Interrupt. If RMWUPE = 1, a RESUME signal is sent in USB bus.
1	RCVD RSM	W	Received Resume. The USB Hardware sets this bit when a USB resume signaling is detected at its port.
0	SUSP	W	Suspend. The USB Hardware sets this bit when it detects no SOF for 3 ms. The USB macro enters in SUSPEND mode, the processor has to go in SLEEP mode.

SPRSIE: Suspend/Resume Interrupt Enable Register

Register Address: 0x0F9

Default State: xxxxx000b

Bit	Field	USB Hardware	Description
7	Reserved		
6	Reserved		
5	Reserved		
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

UISR – USB Interrupt Status Register

Register Address: 0x0F7

Default State: 0x00

Bit	Field	USB Hardware	Description
7	Reserved		
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

The function interrupt bits will be set by the USB Hardware whenever the following bits in the corresponding endpoint's Control and Status registers are modified by the USB Hardware:

1. RX OUT Packet is set (Control and OUT endpoints)
2. TX Packet Ready is cleared (Control and IN endpoints)
3. RX SETUP is set (Control endpoints only)
4. TX Complete is set (Control endpoints only)

UIAR – USB Interrupt Acknowledge Register

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.

Register Address: 0x0F5

Default State: 0x00

Bit	Field	USB Hardware	Description
7	Reserved		
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

UIER – USB Interrupt Enable Register

Register Address: 0x0F3

Default State: 0x00

The bits in this register have the following meaning:

1 = enable interrupt

0 = disable interrupt

Bit	Field	USB Hardware	Description
7	SOF IE	R	Enable SOF 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

Function Address Register

The FIU contains an address register that contains the function address assigned by the host. This Function Address Register must be programmed by the processor once it has received a SET_ADDRESS command from the host and has 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 Hardware. Once this bit is set, the USB Hardware passes packets to and from the host.

Register Address: 0x0F2

Default State: 0x00

Bit	Field	USB Hardware	Description
7	FEN	R	Function Enable
6 - 0	FADD[6:0]	R	Function Address

ENDPPGPG: Endpoint Ping-pong Enable Register

Register Address: 0x0F2

Default State: 0x00

Bit	Field	USB Hardware	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

Endpoint Control Register

Register Address: 0x0EF, ENDP0_CNTR Endpoint0
0x0EE, ENDP1_CNTR Endpoint1
0x0ED, ENDP2_CNTR Endpoint2
0x0EC, ENDP3_CNTR Endpoint3
0x0EB, ENDP4_CNTR Endpoint4
0x0EA, ENDP5_CNTR Endpoint5
0x0E9, ENDP6_CNTR Endpoint6

Default State: 0x000000b

Bit	Field	USB Hardware	Description
7	EPEDS	R	Endpoint Enable/Disable (0 = Disable Endpoint, 1 = Enable Endpoint)
6	Reserved	R	Reserved
4 - 5	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

Endpoint Control and Status Register

Register Address: 0x0DF, FCSR0 Endpoint0
0x0DE, FCSR1 Endpoint1
0x0DD, FCSR2 Endpoint2
0x0DC, FCSR3 Endpoint3
0x0DB, FCSR4 Endpoint4
0x0DA, FCSR5 Endpoint5
0x0D9, FCSR6 Endpoint6

Default State: 00000000b

Bit	Field	USB Hardware	Description
7	Control Direction	R	Set by the processor to indicate to the USB Hardware the direction of a control transfer. 0 = control write (no data stage) 1 = control read This bit is used by control endpoints only.
6	Data End	R	Indicate 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.
5	Force Stall	R	Set by the processor to indicate a stalled endpoint. The USB Hardware will send a STALL handshake as a response to the next IN or OUT token.
4	TX Packet Ready	R/C	Indicates that the processor has loaded the FIFO with a packet of data. This bit is cleared by USB Hardware after the USB host acknowledges the packet. For ISO endpoints, this bit is cleared unconditionally after the data is sent.
3	Stall Snd	W	The USB Hardware sets this bit after a STALL is sent to the host. Indicates end of data stage for the control endpoint only.
2	RX SETUP	W	The USB Hardware sets this bit when it receives a valid setup packet from the host. This bit is used by control endpoints only.
1	RX OUT Packet	W	Indicates that the USB Hardware has decoded an OUT token and that the data is in the FIFO.
0	TX Complete	R	The USB Hardware sets this bit to indicate to a control endpoint that it has received an ACK handshake from the host.

- **Control Direction**

This bit is used by control endpoints only and is used by firware to indicate the direction of a control transfer. It is written by the firware after it receives a RX SETUP Interrupt. The USB Hardware uses this bit to determine the status phase of a control transfer.

- **Data End**

This bit is used only by control endpoints together with bit 1 (TX Packet Ready) to signal the USB Hardware to go to the STATUS phase after the packet currently residing in the FIFO is transmitted.

After the USB Hardware completes the STATUS phase, it will interrupt the processor without clearing this bit.

Caution:

Since 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. That is why Tx_Packet Ready should be set before Data_End.

- **Force Stall**

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.

- **Stall Snd**

The USB Hardware sets this bit after a STALL has been sent. The firmware uses this bit when responding to a USB GetStatus request.

- **TX Packet Ready**

This bit is used for the following operations:

1. Control read transactions by a control endpoint
2. 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”
3. 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 zero length. For a control endpoint, the processor should write to the FIFO only while bit 6 (TX Packet Requested) is set. The USB Hardware clears this bit after it receives an ACK. If the interrupt is enabled, clearing this bit by the USB Hardware causes an interrupt to the processor.

- **RX SETUP**

This bit is used by control endpoints only to signal to the processor that the USB Hardware has received a valid SETUP packet and that the data portion of the packet is stored in the FIFO. The USB Hardware will clear all other bits in this register and will set RX SETUP. If the corresponding interrupt is enabled, the processor will be interrupted when RX SETUP is set. After the completion of reading the data from the FIFO, the firmware should clear this bit.

- **RX OUT Packet**

The USB Hardware sets this bit after it has stored the data of an OUT transaction in the FIFO. While this bit is set, the USB Hardware will NAK all OUT tokens. For control endpoints only, bit 7 of this register, Enable Control Write, has to be set for the USB Hardware to accept the OUT data. The USB Hardware 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:

1. Control write transactions by a control endpoint.
2. OUT transaction with DATA1 PID to complete the status phase of a control endpoint.
3. 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 are read.

- **TX Complete**

This bit is used by USB Hardware 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:

1. Control read data stage
2. Status stage without data stage
3. Status stage after a control write transaction

FIFO Data Registers

These 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 through this register.

Register Address: 0x0CF, FDR0 Function Endpoint0
0x0CE, FDR1 Function Endpoint1
0x0CD, FDR2 Function Endpoint2
0x0CC, FDR3 Function Endpoint3
0x0CB, FDR4 Function Endpoint4
0x0CA, FDR5 Function Endpoint5
0x0C9, FDR6 Function Endpoint6

Default State: 00000000b

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

Byte Count Registers

Each endpoint has a register that stores the number of bytes to be sent or that was received by the USB Hardware. The maximum data packet supported is 1024 bytes in length for isochronous endpoints.

Register Address: 0x0BF, FBYTE_CNT0_L Endpoint0
0x0BE, FBYTE_CNT1_L Endpoint1
0x0BD, FBYTE_CNT2_L Endpoint2
0x0BC, FBYTE_CNT3_L Endpoint3
0x0BB, FBYTE_CNT4_L Endpoint4
0x0BA, FBYTE_CNT5_L Endpoint5
0x0B9, FBYTE_CNT6_L Endpoint6
0x0B8, FBYTE_CNT7_L Endpoint7
0x0AF, FBYTE_CNT0_H Endpoint0
0x0AE, FBYTE_CNT1_H Endpoint1
0x0AD, FBYTE_CNT2_H Endpoint2
0x0AC, FBYTE_CNT3_H Endpoint3
0x0AB, FBYTE_CNT4_H Endpoin4
0x0AA, FBYTE_CNT5_H Endpoint5
0x0A9, FBYTE_CNT6_H Endpoint6

Default State: 00000000b

FBYTE_CNT_L

Bit	Field	USB Hardware	Description
7 - 0	BYTCT[7:0]	R/W	Length of data packet in FIFO

FBYTE_CNT_H

Bit	Field	USB Hardware	Description
7 - 3	Reserved	Reserved	Reserved
2 - 0	BYTCT[10:8]	R/W	Length of data packet in FIFO
2 - 0	ENDSZ[10:8]	R	Endpoint Size [10:8]

Electrical Characteristics

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to 7.0V
Maximum Operating Voltage	3.7V
DC Output Current	16.0 mA

*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.

DC Characteristics

The values shown in this table are valid for TA = 0°C to 85°C, V_{CC} = 3.3V unless otherwise noted.

Power Supply

Symbol	Parameter	Condition	Min	Max	Unit
V_{CC}	Power Supply			3.3	V
I_{CC}	Supply Current			50.0	mA
I_{CCS}	Suspended Device Current			200.0	µA

USB Signals: DP, DM

Symbol	Parameter	Condition	Min	Max	Unit
I_{LO}	High-Z Data Line Leakage	0V < V_{IN} < 3.3V	-10.0	+10.0	µA
V_{DI}	Differential Input Sensitivity	DPx and DMx	0.2		V
V_{CM}	Differential Common Mode Range		0.8	2.5	V
V_{SE}	Single-ended Receiver Threshold		0.8	2.0	V
V_{CRS}	Output Signal Crossover	Except first transition from idle state	1.3	2.0	V
V_{OL1}	Static Output Low	RL of 15 kOhm to 3.6V		0.3	V
V_{OH1}	Static Output High	RL of 15 kOhm to GND			

Oscillator Signals: OSC1, OSC2

Symbol	Parameter	Condition	Min	Max	Unit
V_{LH}	OSC1 Switching Level		0.47	1.20	V
V_{HL}	OSC1 Switching Level		0.67	1.44	V
C_{X1}	Input Capacitance, OSC1			9.0	pF
C_{X2}	Output Capacitance, OSC2[1]			9.0	pF
C_{12}	OSC1/2 Capacitance			1.0	pF
t_{SU}	Start-up Time	6 MHz, fundamental		2.0	ms

Note: 1. OSC2 must not be used to drive other circuitry.

AC Characteristics

DP, DM Driver Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
t_R	Rise Time	CL = 50 pF	4.0	20.0	ns
t_F	Fall Time	CL = 50 pF	4.0	20.0	ns
t_{RFM}	t_R/t_F Matching		90.0	110.0	%
V_{CRS}	Output Signal Crossover	Except first transition from idle state	1.3	2.0	V
Z_{DRV}(1)	Driver Output Resistance	Steady-state drive	29.0	44.0	W

Note: 1. With external 27W series resistor.

DP, DM Data Source Timings

Symbol	Parameter	Condition	Min	Max	Unit
t_{DRATE}	Full-speed Data Rate	Average Bit Rate	11.97	12.03	Mbps
t_{FRAME}	Frame Interval		0.9995	1.0005	ms
t_{DJ1}	Source Differential Driver Jitter to Next Transition for Paired Transitions		-3.5	3.5	ns
t_{DJ2}			-4.0	4.0	ns
t_{FDEOP}	Source Jitter for Differential Transition to SE0 Transition		-2.0	5.0	ns
t_{FEOPT}	Source SE0 Interval of EOP		160.0	175.0	ns
t_{FEOPR}	Receiver SE0 Interval of EOP		82.0		ns
t_{JR1}	Receiver Data Jitter Tolerance to Next Transition for Paired Transitions		-18.5	18.5	ns
t_{JR2}			-9.0	9.0	ns
t_{FST}	Width of SE0 Interval during Differential Transition			14.0	ns

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
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 Product Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-7658-3000
FAX (33) 4-7658-3480

Atmel Heilbronn
Theresienstrasse 2
POB 3535
D-74025 Heilbronn, Germany
TEL (49) 71 31 67 25 94
FAX (49) 71 31 67 24 23

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 0 2 40 18 18 18
FAX (33) 0 2 40 18 19 60

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Atmel Smart Card ICs
Scottish Enterprise Technology Park
East Kilbride, Scotland G75 0QR
TEL (44) 1355-357-000
FAX (44) 1355-242-743

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

1643B–09/01/xM