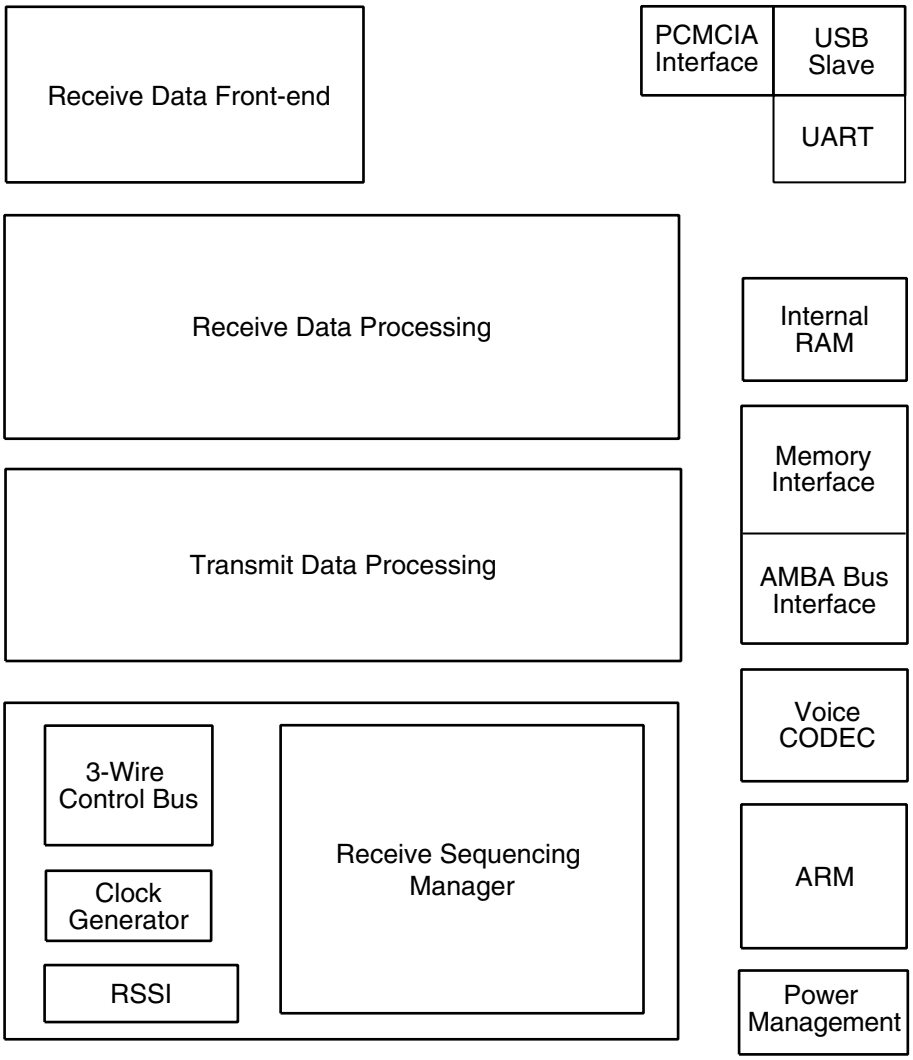
Features

- **Implements Bluetooth™ Specification on Short Distance Wireless Communication in 2.4 GHz ISM Band**
- **Provides 1 Mbps Aggregate Bit Rate**
- **Supports Frequency Hopping Spread Spectrum Physical-layer Interface to Dedicated Transceiver with Frequency Hopping Algorithm Implemented in Hardware**
- **Provides Baseband Functions in Hardware which Implement Bluetooth Low-level Bit Processing Such as Forward Error Correction (FEC), Header Error Check (HEC) and CRC Generation/Checking and Encryption/Decryption**
- **Integrated ARM7TDMI® RISC Processor**
- **Glueless SRAM Interface, Supporting Up to 256K Bytes of Memory**
- **Glueless Flash Memory Interface, Supporting Up to 256K Bytes of Nonvolatile Memory**
- **Glueless PCMCIA Bus Interface Conforming to PC Card Standard – Feb. 1995**
- **USB Interface Conforming to Universal Serial Bus Standard Version 1.1**
- **16550 UART Core Offering 32-byte Receive FIFO and Programmable Baud Rate**
- **Programmable 8/16-bit Wide External Memory Interface**
- **Supports Multiple Reference Clock Frequencies (13.000, 14.400, 16.800, 19.440 MHz)**
- **176-lead LQFP**
- **3.3V Supply**

System Level Block Diagram



Single Chip Bluetooth™ Controller

AT76C551

Rev. 1612D–08/01

Overview

The AT76C551 is a single chip controller providing the functionality for high data rate, short distance wireless communications in the free ISM band. In conjunction with a 2.4 GHz transceiver, it provides a cost effective networking solution for a wide range of digital communication devices and computer peripherals. Integration is simplified due to the incorporation of three different interfaces: USB and 16550 UART compatible interfaces and a PCMCIA interface conforming to the PC Card 95 specification. Additionally, a voice coding/decoding module is provided.

The AT76C551 is comprised of a baseband processor. This processor carries out all bit-level processing after modulation/demodulation of the Bluetooth bitstream. It controls the transceiver and dedicated voice coding/decoding. The AT76C551 has an ARM7TDMI processor core with support for internal and external memory, as well as the interface core logic.

The powerful RISC processor in the ARM7TDMI carries out all but the low level baseband functions.

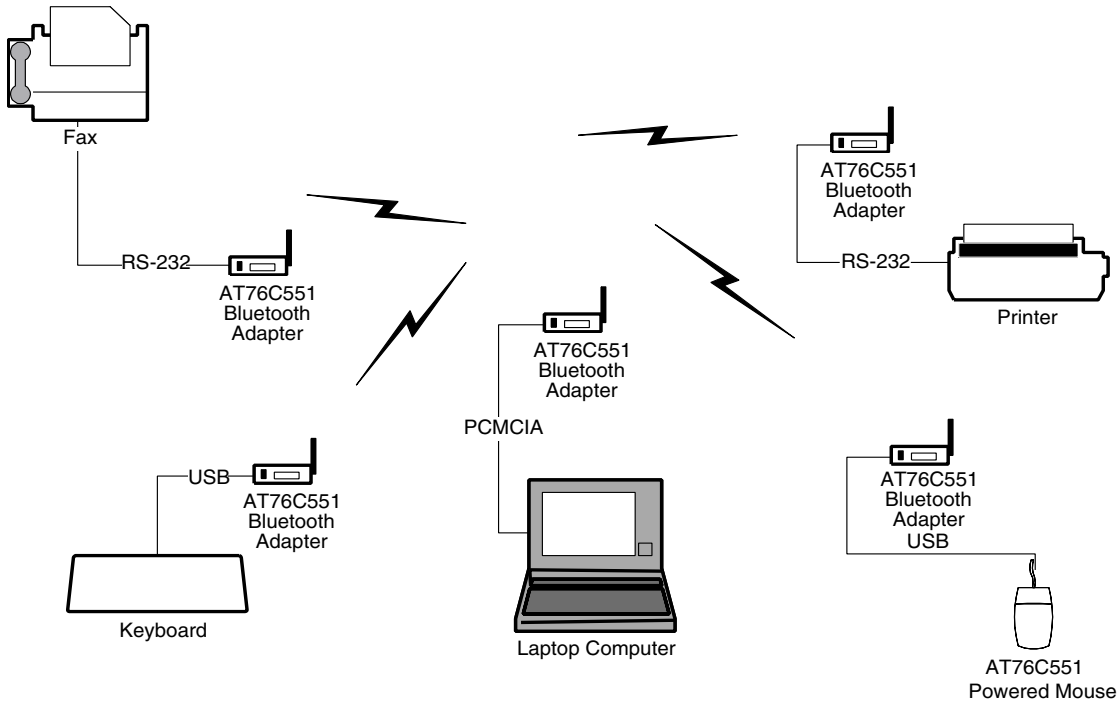
Applications

AT76C551 can be used in applications where fast short range communication is required between portable devices such as mobile phones and digital peripherals.

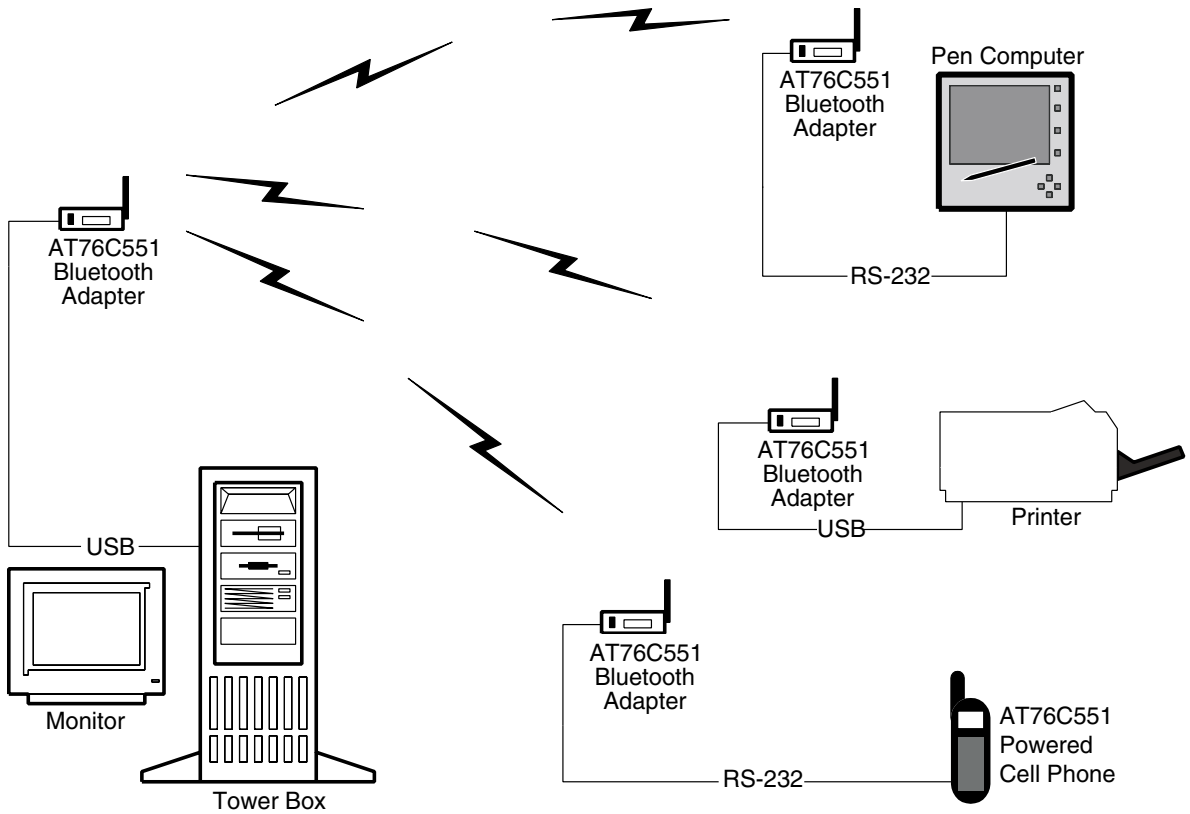
Typical usages would include:

- Wireless network cards
- Mobile phones
- Laptop and desktop computers
- Pagers
- Digital cameras
- PDAs
- Wireless computer peripherals (printers, etc.)

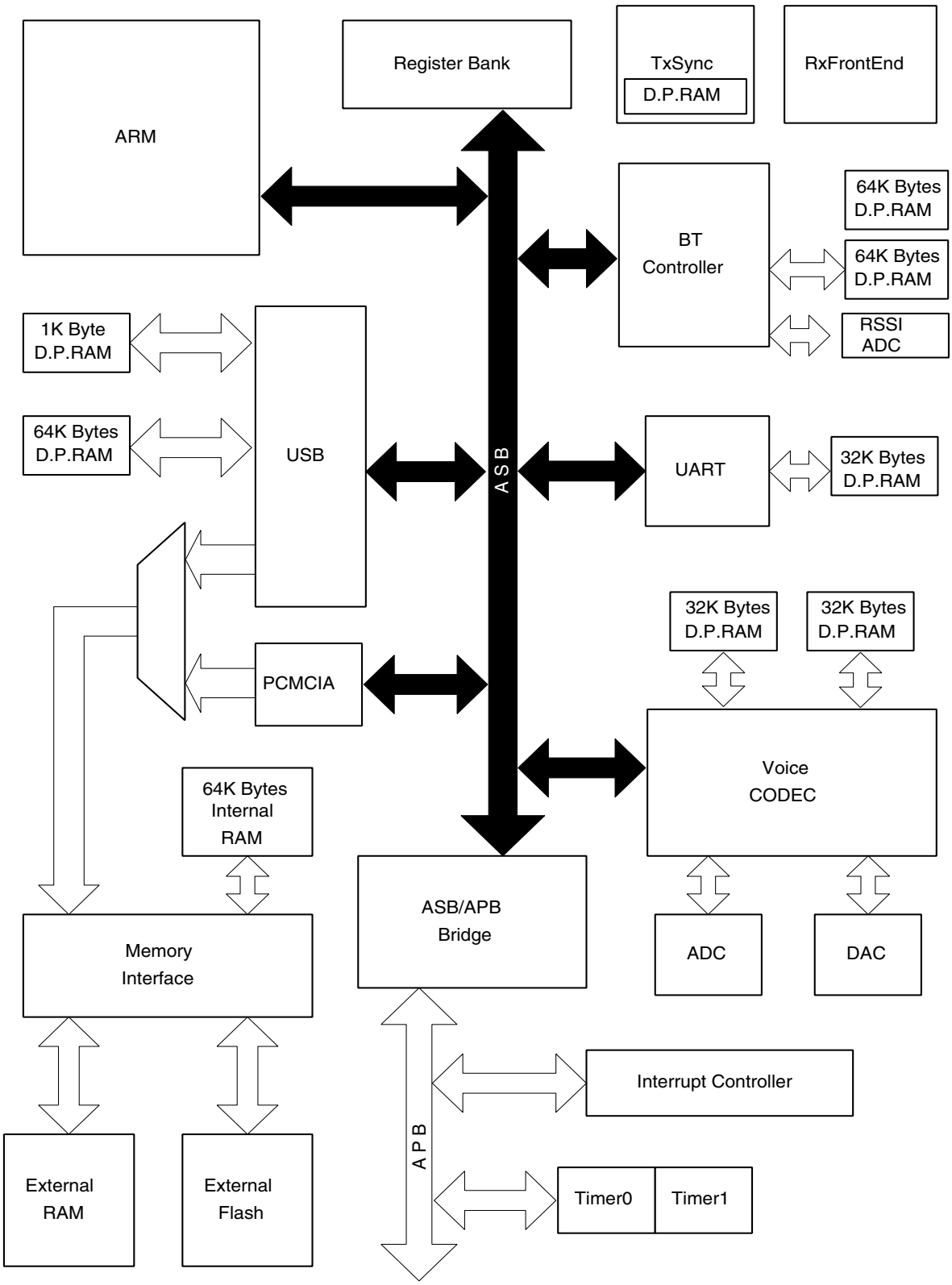
Typical AT76C551 Home Application



Typical AT76C551 Mobile Application



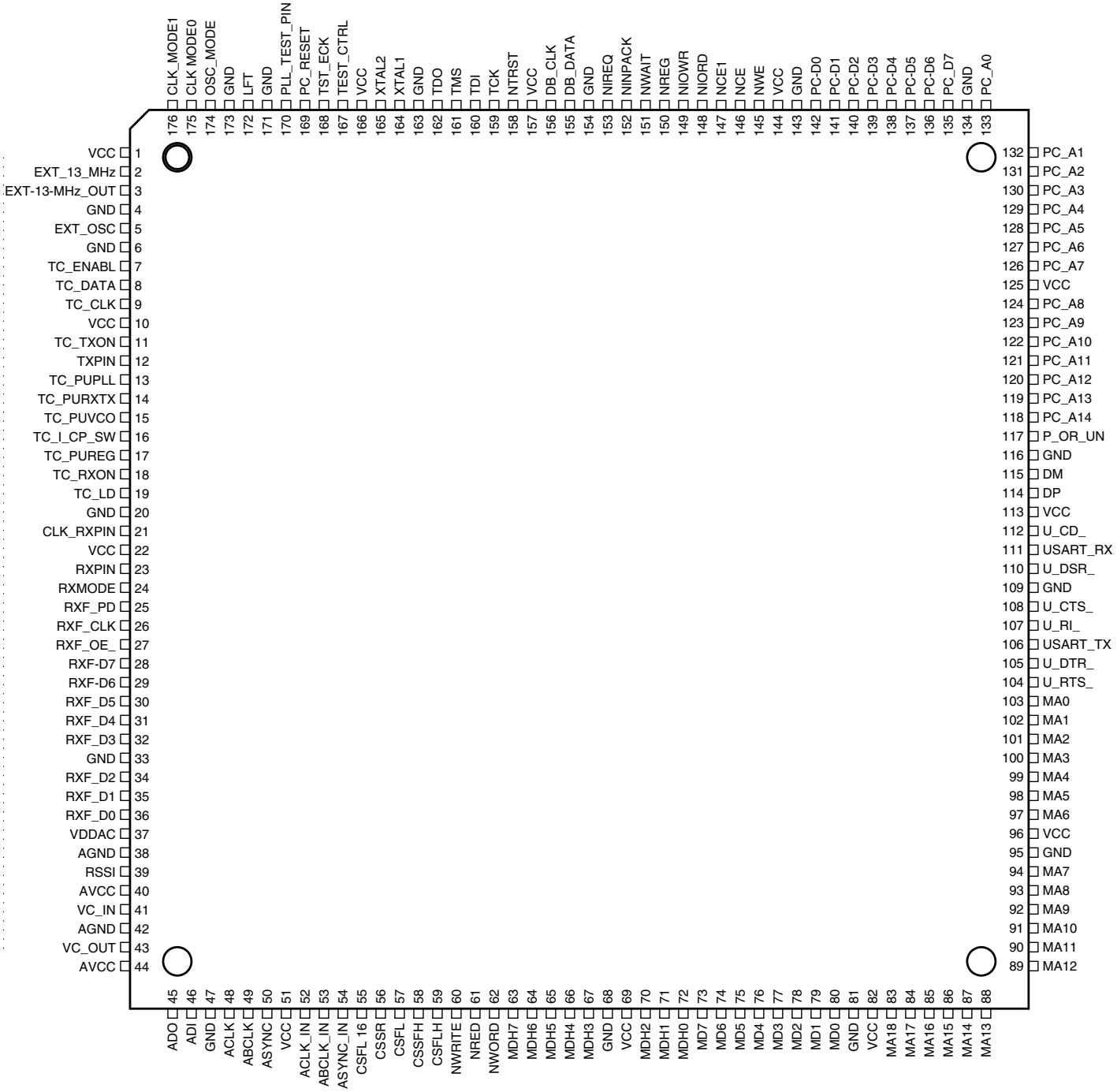
Functional Diagram



Pinout and Package Options

The AT76C551 controller will be available in three different packages, each will have the same basic functionality but with a different system interface (PCMCIA 8-bit, full-speed USB, extended speed RS-232). The prototype version comes in a LQFP-176 package and supports the three different interfaces simultaneously.

Pinout of the Prototype Version



Pin Summary – Pin Assignment in Numerical Order

Pin #	Pin Name	Type
1	VCC	Digital Supply
2	EXT_13_MHz	I
3	EXT_13_MHz_OUT	O
4	GND	Digital Ground
5	EXT_OSC	I
6	GND	Digital Ground
7	TC_ENABL	O
8	TC_DATA	O
9	TC_CLK	O
10	VCC	Digital Supply
11	TC_TXON	O
12	TXPIN	O
13	TC_PUPLL	O
14	TC_PURXTX	O
15	TC_PUVCO	O
16	TC_I_CP_SW	O
17	TC_PUREG	O
18	TC_RXON	O
19	TC_LD	I
20	GND	Digital Ground
21	CLK_PXPIN	I
22	VCC	Digital Supply
23	RXPIN	I
24	RXMODE	I
25	RXF_PD	O
26	RXF_CLK	O
27	RXF_OE_	O
28	RXF_D7	I
29	RXF_D6	I
30	RXF_D5	I
31	RXF_D4	I
32	RXF_D3	I
33	GND	Digital Ground
34	RXF_D2	I
35	RXF_D1	I
36	RXF_D0	I
37	VDDAC	
38	AGND	Analog Ground
39	RSSI	I
40	AVCC	
41	VC_IN	LOG 1
42	AGND	Analog Ground
43	VC_OUT	LOG O
44	AVCC	
45	ADO	O
46	ADI	I
47	GND	Digital Ground
48	ACLK	O
49	ABCLK	O
50	ASYNC	O
51	VCC	Digital Supply
52	ACLK_IN	I
53	ABCLK_IN	I
54	ASYNC_IN	I
55	CSFL16	O
56	CSSR	O
57	CSFL	O
58	CSSFH	
59	CSFLH	O
60	NWRITE	O
61	NRED	
62	NWORD	I
63	MEM_DATAH7	B
64	MEM_DATAH6	B
65	MEM_DATAH5	B
66	MEM_DATAH4	B
67	MEM_DATAH3	B
68	GND	Digital Ground
69	VCC	Digital Supply
70	MEM_DATAH2	B
71	MEM_DATAH1	B
72	MEM_DATAH0	B
73	MEM_DATA7	B
74	MEM_DATA6	B
75	MEM_DATA5	B
76	MEM_DATA4	B
77	MEM_DATA3	B
78	MEM_DATA2	B
79	MEM_DATA1	B
80	MEM_DATA0	B
81	GND	Digital Ground
82	VCC	Digital Supply
83	MEM_ADDR18	O
84	MEM_ADDR17	O
85	MEM_ADDR16	O
86	MEM_ADDR15	O
87	MEM_ADDR14	O
88	MEM_ADDR13	O
89	MEM_ADDR12	O
90	MEM_ADDR11	O
91	MEM_ADDR10	O
92	MEM_ADDR9	O
93	MEM_ADDR8	O
94	MEM_ADDR7	O
95	GND	Digital Ground
96	VCC	Digital Supply
97	MEM_ADDR6	O
98	MEM_ADDR5	O
99	MEM_ADDR4	O
100	MEM_ADDR3	O
101	MEM_ADDR2	O
102	MEM_ADDR1	O
103	MEM_ADDR0	O
104	U_RTS_	O
105	U_DTR_	O
106	USART_TX	O
107	U_RI_	I
108	U_CTS_	I
109	GND	Digital Ground
110	U_DSR_	I
111	USART_RX	I

Pin Summary – Pin Assignment in Numerical Order (Continued)

Pin #	Pin Name	Type
112	U_CD_	I
113	VCC	Digital Supply
114	DP	B
115	DM	B
116	GND	Digital Ground
117	P_OR_UN	I
118	PC_A14	I
119	PC_A13	I
120	PC_A12	I
121	PC_A11	I
122	PC_A10	I
123	PC_A9	I
124	PC_A8	I
125	VCC	Digital Supply
126	PC_A7	I
127	PC_A6	I
128	PC_A5	I
129	PC_A4	I
130	PC_A3	I
131	PC_A2	I
132	PC_A1	I
133	PC_A0	I

Pin #	Pin Name	Type
134	GND	Digital Ground
135	PC_D7	B
136	PC_D6	B
137	PC_D5	B
138	PC_D4	B
139	PC_D3	B
140	PC_D2	B
141	PC_D1	B
142	PC_D0	B
143	GND	Digital Ground
144	VCC	Digital Supply
145	NWE	I
146	NCE	
147	NCE1	I
148	NIORD	I
149	NIOWR	I
150	NREG	I
151	NWAIT	O
152	NINPACK	O
153	NIREQ	
154	GND	Digital Ground
155	DB_DATA	O

Pin #	Pin Name	Type
156	DB_CLK	O
157	VCC	Digital Supply
158	NTRST	I
159	TCK	I
160	TDI	I
161	TMS	I
162	TDO	I
163	GND	Digital Ground
164	XTAL1	I
165	XTAL2	I
166	VCC	Digital Supply
167	TEST_CTRL	I
168	TST_ECK	I
169	PC_RESET	I
170	PLL_TEST_PIN	I
171	GND	Digital Ground
172	LFT	
173	GND	Digital Ground
174	OSC_MODE	
175	CLK_MODE0	
176	CLK_MODE1	

Pin Summary – Pin Assignment in Alphabetical Order

Pin #	Pin Name	Type
49	ABCLK	O
53	ABCLK_IN	I
48	ACLK	O
52	ACLK_IN	I
46	ADI	I
45	ADO	O
38	AGND	Analog Ground
42	AGND	Analog Ground
50	ASYNC	O
54	ASYNC_IN	I
40	AVCC	
44	AVCC	
175	CLK_MODE0	
176	CLK_MODE1	
21	CLK_PXPIN	I
57	CSFL	O
55	CSFL16	O
59	CSFLH	O
58	CSSFH	
56	CSSR	O
156	DB_CLK	O
155	DB_DATA	O
115	DM	B
114	DP	B
2	EXT_13_MHz	I
3	EXT_13_MHz_OUT	O
5	EXT_OSC	I
4	GND	Digital Ground
6	GND	Digital Ground
20	GND	Digital Ground
33	GND	Digital Ground
116	GND	Digital Ground
47	GND	Digital Ground
68	GND	Digital Ground
134	GND	Digital Ground
143	GND	Digital Ground
154	GND	Digital Ground

Pin #	Pin Name	Type
81	GND	Digital Ground
95	GND	Digital Ground
109	GND	Digital Ground
163	GND	Digital Ground
171	GND	Digital Ground
173	GND	Digital Ground
172	LFT	
103	MEM_ADDR0	O
102	MEM_ADDR1	O
91	MEM_ADDR10	O
90	MEM_ADDR11	O
89	MEM_ADDR12	O
88	MEM_ADDR13	O
87	MEM_ADDR14	O
86	MEM_ADDR15	O
85	MEM_ADDR16	O
84	MEM_ADDR17	O
83	MEM_ADDR18	O
101	MEM_ADDR2	O
100	MEM_ADDR3	O
99	MEM_ADDR4	O
98	MEM_ADDR5	O
97	MEM_ADDR6	O
94	MEM_ADDR7	O
93	MEM_ADDR8	O
92	MEM_ADDR9	O
80	MEM_DATA0	B
79	MEM_DATA1	B
78	MEM_DATA2	B
77	MEM_DATA3	B
76	MEM_DATA4	B
75	MEM_DATA5	B
74	MEM_DATA6	B
73	MEM_DATA7	B
72	MEM_DATAH0	B
71	MEM_DATAH1	B
70	MEM_DATAH2	B

Pin #	Pin Name	Type
67	MEM_DATAH3	B
66	MEM_DATAH4	B
65	MEM_DATAH5	B
64	MEM_DATAH6	B
63	MEM_DATAH7	B
146	NCE	
147	NCE1	I
152	NINPACK	O
148	NIORD	I
149	NIOWR	I
153	NIREQ	
61	NRED	
150	NREQ	I
158	NTRST	I
151	NWAIT	O
145	NWE	I
62	NWORD	I
60	NWRITE	O
174	OSC_MODE	
117	P_OR_UN	I
133	PC_A0	I
132	PC_A1	I
122	PC_A10	I
121	PC_A11	I
120	PC_A12	I
119	PC_A13	I
118	PC_A14	I
131	PC_A2	I
130	PC_A3	I
129	PC_A4	I
128	PC_A5	I
127	PC_A6	I
126	PC_A7	I
124	PC_A8	I
123	PC_A9	I
142	PC_D0	B
141	PC_D1	B

Pin Summary – Pin Assignment in Alphabetical Order (Continued)

Pin #	Pin Name	Type
140	PC_D2	B
139	PC_D3	B
138	PC_D4	B
137	PC_D5	B
136	PC_D6	B
135	PC_D7	B
169	PC_RESET	I
170	PLL_TEST_PIN	I
39	RSSI	I
26	RXF_CLK	O
36	RXF_D0	I
35	RXF_D1	I
34	RXF_D2	I
32	RXF_D3	I
31	RXF_D4	I
30	RXF_D5	I
29	RXF_D6	I
28	RXF_D7	I
27	RXF_OE_	O
25	RXF_PD	O
24	RXMODE	I
23	RXPIN	I

Pin #	Pin Name	Type
9	TC_CLK	O
8	TC_DATA	O
7	TC_ENABL	O
16	TC_I_CP_SW	O
19	TC_LD	I
13	TC_PUPLL	O
17	TC_PUREG	O
14	TC_PURXTX	O
15	TC_PUVCO	O
18	TC_RXON	O
11	TC_TXON	O
159	TCK	I
160	TDI	I
162	TDO	I
167	TEST_CTRL	I
161	TMS	I
168	TST_ECK	I
12	TXPIN	O
112	U_CD_	I
108	U_CTS_	I
110	U_DSR_	I
105	U_DTR_	O

Pin #	Pin Name	Type
107	U_RI_	I
104	U_RTS_	O
111	USART_RX	I
106	USART_TX	O
41	VC_IN	LOG 1
43	VC_OUT	LOG O
1	VCC	Digital Supply
10	VCC	Digital Supply
22	VCC	Digital Supply
113	VCC	Digital Supply
125	VCC	Digital Supply
51	VCC	Digital Supply
69	VCC	Digital Supply
144	VCC	Digital Supply
82	VCC	Digital Supply
96	VCC	Digital Supply
157	VCC	Digital Supply
166	VCC	Digital Supply
37	VDDAC	
164	XTAL1	I
165	XTAL2	I

Signal Description – Pin Name Order

I = Input, O = Output, B = Bidirectional, Analog I = Analog Input, Analog O = Analog Output

Pin Name	Type	Description
Supply Pins		
AGND	Power	Analog Ground
AVCC	Power	Analog Supply
GND	Power	Digital Ground
VCC	Power	Digital Supply
Global Pin		
PC_RESET	I	Global Reset pin
Oscillator Pins		
XTAL1	I	Crystal oscillator input
XATL2	O	Crystal oscillator output
EXT_13_MHZ	I	13 MHz input clock
EXT_13_MHZ_OUT	O	13 MHz output clock
EXT_OSC	I	External oscillator input
CLK_MODE1, CLK_MODE0	I	Clock Frequency Configuration
OSC_MODE	I	Internal/Extenal Clock Selector
LFT	PLL Filter	PLL loop filter pin
Memory Interface Pins		
CSFL16	O	Chip select when 16-bit Flash is used
CSSR	O	Chip select for low byte 8-bit SRAM
CSFL	O	Chip select for low byte Flash
CSSRH	O	Chip select for high byte SRAM
CSFLH	O	Chip select for high byte Flash
NWRITE	O	Memory write
NREAD	O	Memory read
NWORD	I	Selects between 8 and 16 memory access
MEM_ADDR [18:0]	O	Memory Address Bus – Signals MEM_ADDR [18:0] are address-bus output lines of Flash and SRAM.
MEM_DATA [7:0]	B	Memory Data Bus – Signals MEM_DATA [7:0] are the bi-directional data bus lines for the SRAM and Flash memory.
MEM_DATAH [7:0]	B	Memory Data Bus – Signals MEM_DATAH [7:0] are the bi-directional data bus lines for the SRAM and Flash memory. These lines are used during 16-bit memory configuration.
P_OR_UN	I	Selects whether PCMCIA or USB will have direct access to the memories. When “1” PCMCIA has direct access while when “0” USB has access

Signal Description – Pin Name Order (Continued)

I = Input, O = Output, B = Bidirectional, Analog I = Analog Input, Analog O = Analog Output

Pin Name	Type	Description
Baseband Interface Pins		
TC_CLK	O	Clock output for transceiver control bus
TC_DATA	O	Data output for transceiver control bus
TC_ENBL	O	Enable output for transceiver control bus
TC_LD	I	Lock detect input
TC_PU_REG	O	Transceiver voltage regulator power up
TC_PU_PLL	O	Power up output for transceiver PLL
TC_PU_VCO	O	VCO power up output
TC_PU_RXTX	O	Power up output for Transmit/Receive sections of transceiver
TC_RX_ON	O	RX section power up control output
TC_TX_ON	O	TX section power up control output
TC_I_CP_SW	O	Transceiver Current Pump external switch signal
RSSI	I	Receive signal strength indicator input from transceiver
TXPIN	O	Transmit data output to transceiver
RXPIN	I	Receive data input from transceiver
CLK_RXPIN	I	Clock driven to internal RxFront module
RX_MODE	I	If “0”, internal clock recovery enabled and data extracted from external A/D. If “1” only BT packet processing carried out. Clock generation and recovery must be carried out externally
RXF_PD	O	Power Down signal for external A/D
RXF_CLK	O	Clock driven to external A/D
RXF_OE_	O	Output enable signal for external A/D
RXF_D[7:0]	I	A/D RXF_D[7:0] output bus
PCMCIA Interface Pins		
PC_A[14:0]	I	Address Bus – Signals A0 through A14 are address-bus input lines. Signal A0 is always used since the data interface is 8 bits wide.
PC_D[7:0]	B	Data Bus – Signals D7 through D0 are the bi-directional data bus for PCMCIA. The most significant bit is D7.
NWE	I	Write Enable – Used to strobe memory write data into the chip from the PCMCIA data bus. Should be deasserted during memory read cycles. It is used for both Common memory and Attribute memory accesses.
NOE	I	Output Enable – Active low output enable signal. It is used to gate memory read data from the BT device onto PCMCIA data bus. It is used for Common memory accesses and Attribute memory Accesses.
NCE1	I	Card Enable – Used to enable even-numbered word address bytes. A0 is used to select between the even and odd number bytes within the addressed word

Signal Description – Pin Name Order (Continued)

I = Input, O = Output, B = Bidirectional, Analog I = Analog Input, Analog O = Analog Output

Pin Name	Type	Description
NIORD	I	I/O Read – Asserted by the host system to indicate to BT that a read from the I/O address space is required. The chip will not respond until it has been configured for I/O operation by the system.
NIOWR	I	I/O Write – Asserted by the host system to indicate to the chip that a write to its I/O address space is required. The device will not respond until it has been configured for I/O operation by the system.
NREG	I	Attribute Memory Select – Driven by the host to select between Attribute memory or I/O space (REG asserted) and Common memory (REG deasserted) in the device and the PCMCIA card.
NWAIT	O	Extend Bus Cycle – This signal is asserted by the device to delay completion of the access cycle currently in progress.
NINPACK	O	Input Acknowledge – It is asserted when the BT device is selected and can respond to an I/O read cycle at the address currently applied on the address bus. It is used by the host to control the enable of any input buffer between the card and the CPU. It will be inactive during card configuration.
NIREQ	O	Interrupt Request – Asserted by the chip to indicate to the host that software service should take place.
UART Interface Pins		
U_CTS_	I	Clear To Send
U_DSR_	I	Data Set Ready
U_DTR_	O	Data Terminal Ready
U_RI_	I	Ring Indicator
U_RTS_	O	Request To Send
USART_RX	I	Serial Input Port
USART_TX	O	Serial Output Port
U_CD_	I	Carrier Detect
USB Interface Pins		
DP	B	Upstream Plus USB I/O. This pin should be connected to CEXT through an external 1.5 kΩ pull-up resistor. DPLUS and DMINUS form the differential signal pin pairs connected to the Host Controller or an upstream Hub.
DM	B	Upstream Minus USB I/O
Analog Voice CODEC Pins		
VC_IN	LOG I	Voice Input
VC_OUT	LOG O	Voice Output
Digital Voice CODEC Pins		
ADO	O	Data Transmit
ADI	I	Data Receive
ACLK	O	Master Clock – Out
ABCLK	O	Bit Clock – Out

Signal Description – Pin Name Order (Continued)

I = Input, O = Output, B = Bidirectional, Analog I = Analog Input, Analog O = Analog Output

Pin Name	Type	Description
ASYNC	O	Frame Sync – Out
ACLK_IN	I	Master Clock – In
ABCLK_IN	I	Bit Clock – In
ASYNC_IN	I	Frame Sync – In
JTAG Pins		
DB_DATA	O	Debug data port
DB_CLK	O	Debug clock port
NTRST	I	JTAG reset input
TCK	I	JATG clock
TDI	I	JTAG data input
TDO	I	JTAG data output
TMS	I	JTAG master select input
TEST Pins		
TEST_CTRL	I	For production test
TEST_ECK	I	For production test
PLL_TEST_PIN	I	For production test.

Functional Description – Pin Name Order

Name	Type	Description
Supply Pins		
AGND	Power	Analog Ground – used by the RSSI ADC, the Voice CODEC ADC and DAC
AVCC	Power	Analog Supply – used by the RSSI ADC, the Voice CODEC ADC and DAC
GND	Power	Digital Ground
VCC	Power	Digital Supply
Global Pin		
PC_RESET	I	Global Reset Pin
Oscillator Pins		
XTAL1	I	Crystal oscillator input
XATL2	O	Crystal oscillator output
EXT_13_MHZ	I	13 MHz input clock
EXT_13_MHZ_OUT	O	13 MHz output clock
EXT_OSC	I	External oscillator input
CLK_MODE1		Clock Frequency Configuration (see “Support for Different Operating Frequencies”)
CLK_MODE0	I	Clock Frequency Configuration (see “Support for Different Operating Frequencies”)
OSC_MODE	I	Selects if the PLL will use the internal clock oscillator connected to the XTAL1, XTAL2 pins or an external clock oscillator connected to the EXT_OSC pin.
LFT	PLL Filter	PLL loop filter pin
Memory Interface Pins		
CSFL16	O	Chip Select when 16-bit Flash is used
CSSR	O	Chip select for low byte 8-bit SRAM
CSFL	O	Chip select for low byte Flash
CSSRH	O	Chip select for high byte SRAM
CSFLH	O	Chip select for high byte Flash
NWRITE	O	Memory write
NREAD	O	Memory read
NWORD	I	Selects between 8 and 16 memory access
MEM_ADDR [18:0]	O	Memory Address Bus – Signals MEM_ADDR [18:0] are address-bus output lines of Flash and SRAM.
MEM_DATA [7:0]	B	Memory Data Bus – Signals MEM_DATA [7:0] are the bi-directional data bus lines for the SRAM and Flash memory.
MEM_DATAH [7:0]	B	Memory Data Bus – Signals MEM_DATAH [7:0] are the bi-directional data bus lines for the SRAM and Flash memory. These lines are used during 16-bit memory configuration
P_OR_UN	I	Selects whether PCMCIA or USB will have direct access to the memories When “1” PCMCIA has direct access while when “0” USB has access

Functional Description – Pin Name Order (Continued)

Name	Type	Description
Baseband Interface Pins		
TC_CLK	O	Clock output for transceiver control bus
TC_DATA	O	Data output for transceiver control bus
TC_ENBL	O	Enable output for transceiver control bus
TC_LD	I	Lock detect input
TC_PUREG	O	Transceiver voltage regulator power up
TC_PUPLL	O	Power up output for transceiver PLL
TC_PUVCO	O	VCO power up output
TC_PURXTX	O	Power up output for Transmit/Receive sections of transceiver
TC_RXON	O	RX section power up control output
TC_TXON	O	TX section power up control output
TC_I_CP_SW	O	Transceiver Current Pump external switch signal
RSSI	I	Receive signal strength indicator input from transceiver
TXPIN	O	Transmit data output to transceiver
RXPIN	I	Receive data input from transceiver
CLK_RXPIN	I	Clock driven to internal RxFront module
RXMODE	I	If “0”, internal clock recovery enabled and data extracted from external A/D. If “1” only BT packet processing carried out. Clock generation and recovery must be carried out externally.
RXF_PD	O	Power Down signal for external A/D
RXF_CLK	O	Clock driven to external A/D
RXF_OE_	O	Output enable signal for external A/D
RXF_D[7:0]	I	A/D RXF_D[7:0] output bus
PCMCIA Interface Pins		
PC_A[14:0]	I	Address Bus – Signals A0 through A14 are address-bus input lines. Signal A0 is always used since the data interface is 8-bits wide.
PC_D[7:0]	B	Data Bus – Signals D7 through D0 are the bi-directional data bus for PCMCIA. The most significant bit is D7.
NWE	I	Write Enable – Used to strobe memory write data into the chip from the PCMCIA data bus. Should be deasserted during memory read cycles. It is used for both Common memory and Attribute memory accesses.
NOE	I	Output Enable – Active low output enable signal. It is used to gate memory read data from the BT device onto PCMCIA data bus. It is used for Common memory accesses and Attribute memory Accesses.
NCE1	I	Card Enable – Used to enable even-numbered word address bytes. A0 is used to select between the even and odd number bytes within the addressed word
NIORD	I	I/O Read – Asserted by the host system to indicate to BT that a read from the I/O address space is required. The chip will not respond until it has been configured for I/O operation by the system.

Functional Description – Pin Name Order (Continued)

Name	Type	Description
NIOWR	I	I/O Write – Asserted by the host system to indicate to the chip that a write to its I/O address space is required. The device will not respond until it has been configured for I/O operation by the system.
NREG	I	Attribute Memory Select – Driven by the host to select between Attribute memory or I/O space (REG asserted) and Common memory (REG deasserted) in the device and the PCMCIA card.
NWAIT	O	Extend Bus Cycle – This signal is asserted by the device to delay completion of the access cycle currently in progress.
NINPACK	O	Input Acknowledge – It is asserted when the BT device is selected and can respond to an I/O read cycle at the address currently applied on the address bus. It is used by the host to control the enable of any input buffer between the card and the CPU. It will be inactive during card configuration.
NIREQ	O	Interrupt Request – Asserted by the chip to indicate to the host that software service should take place.
UART Interface Pins		
U_CTS_	I	Clear to Send
U_DSR_	I	Data Set Ready
U_DTR_	O	Data Terminal Ready
U_RI_	I	Ring Indicator
U_RTS_	O	Request to Send
USART_RX	I	Serial input port
USART_TX	O	Serial output port
U_CD_	I	Carrier Detect
USB Interface Pins		
DP	B	Upstream Plus USB I/O. This pin should be connected to CEXT through an external 1.5 kΩ pull-up resistor. DPLUS and DMINUS form the differential signal pin pairs connected to the Host Controller or an upstream Hub.
DM	B	Upstream Minus USB I/O
Analog Voice CODEC Pins		
VC_IN	LOG I	Voice input
VC_OUT	LOG O	Voice output
Digital Voice CODEC Pins		
ADO	O	Data Transmit
ADI	I	Data Receive
ACLK	O	Master Clock – Out
ABCLK	O	Bit clock – Out
ASYNC	O	Frame Sync – Out
ACLK_IN	I	Master Clock – In
ABCLK_IN	I	Bit Clock – In
ASYNC_IN	I	Frame Sync – In

Functional Description – Pin Name Order (Continued)

Name	Type	Description
JTAG Pins		
DB_DATA	O	Debug data port
DB_CLK	O	Debug clock port
NTRST	I	JTAG reset input
TCK	I	JATG clock
TDI	I	JTAG data input
TDO	I	JTAG data output
TMS	I	JTAG master select input
TEST Pins		
TEST_CTRL	I	For production test
TEST_ECK	I	For production test
PLL_TEST_PIN	I	For production test

Internal Architecture

The AT76C551 chip is based on the ARM7TDMI processor. All modules are connected to the processor through a 32-bit bus including 32 KB internal SRAM. The processor operates at 24 MHz while low speed operation of the ARM® processor and power-down of virtually all interfaces is possible.

ARM Core

The controller contains the ARM7TDMI (ARM7) core, a 32-bit RISC processor. The ARM7 core can execute all the functionally called for by the Bluetooth specification. ARM7 supports two alternative instruction sets:

1. Powerful 32-bit code can be executed by the processor in ARM® operating mode.
2. Thumb®, which stores a subset of 32-bit instructions as compressed 16-bit instructions and decompresses them back to 32 bits upon execution.

Utilization of Thumb mode will exploit full processor power with limited external memory resources.

Note: ARM7TDMI operating mode can be changed at run time with negligible overhead.

Memory Interface

The memory interface interfaces the ARM processor to the 32K internal SRAM and external Flash and SRAM memory that accompanies a complete design. Flash memory contains the Bluetooth and Host Interface firmware and the Card Information Structure (CIS) used by the host PCMCIA subsystem. Either 8-bit or 16-bit Flash memories and 8-bit SRAMs can be used.

The internal 32K SRAM memory accommodates ARM core stack, firmware status variables, structures supporting host/firmware interface and network data buffers. When the host driver software passes network data through the PCMCIA interface unit, the data can be automatically routed to either internal or external memories.

Note: Firmware performance is optimized by permanently storing it in a slow Flash. Upon initialization it is loaded onto faster SRAM to fetch the instructions.

External memory can be either 8-bit or 16-bit wide. External memory width is configured via the NWORD pin. When 8-bit memory is used, up to 512 kilobytes of SRAM and Flash memory is supported. When 16-bit memory is used, up to 1 megabyte of SRAM and Flash memory is supported.

The memory interface supports two ports for efficient use of the memory unit. Each port provides access to all memories independently of the other. A round-robin priority scheme is used when both ports require access to the same memory. One port is dedicated to the ARM interface while the other can be used by either the PCMCIA interface or the USB interface. The selection is determined by the PC_OR_UN pin.

PCMCIA Interface Unit

The PCMCIA interface unit implements a PCMCIA 2.1/JEIDA 4.2 compatible 8-bit wide PC-card front-end interface. The PCMCIA interface unit provides the host with master access to internal/external SRAM and external Flash memory. The PCMCIA interface unit contains a number of general purpose registers to allow configuration and/or status information exchange between the host and Bluetooth firmware. Moreover, the PCMCIA interface unit provides the host and the ARM core with the capability to raise interrupts to each other.

Consequently, the host driver software uses the PCMCIA interface unit to exchange configuration information, to monitor operation, to receive network indications and to transfer network data from/to network data buffers in SRAM.

Baseband Processor

The baseband processor unit carries out the low-level Bluetooth link functions. This unit has been designed to automatically handle many time-critical physical network management tasks used by the Bluetooth link control.

Bluetooth Clock and Internal Hardware Timers

A 28-bit counter running at 3.2 kHz provides the timing signals required for the baseband processing. Also other timers are included which provide general purpose interrupts or determine the timing of specific transceiver programming events.

Frequency Hopping Sequence Generator

The frequency hopping sequence generator chooses the correct hop frequency depending on the Bluetooth clock, the device address and the device mode.

Access Code Generation and Detection

This module generates the correct access code for access to other Bluetooth devices from the corresponding Lower Address Part (LAP) of the utilized Bluetooth Device address. A correlator is used to detect a Bluetooth transmission with a valid access code.

Forward Error Correction (FEC)

Two types of forward error correction are used. The first type is a simple 3-times repetition code where each bit is repeated three times. Majority decision decoding is used in the receiver.

The second type of code is a (15,10) shortened Hamming code. The data is divided into 10-bit blocks. Each block is allocated a 15-bit code word. This code can correct all single-bit errors while it can detect double bit errors in each code word.

Header Error Check (HEC)

Dedicated header error generation and checking is provided to guarantee correct decoding of the important information included in the header. The generator calculates the HEC field in the header of a transmitted Bluetooth packet while the HEC detects corrupted packet headers.

Cyclic Redundancy Generation and Check (CRC)

A 16-bit CRC is used to protect the payload data transmitted using certain types of Bluetooth packets. During transmission, the CRC is automatically generated and appended at the end of the packet. Checking for received BT packets is a fully automated fund of the CRC in the ARM Core. It is a simple matter of checking the corresponding CRC status bit.

Encryption/ Decrytion

Encryption and decryption is provided with the use of a secret key. Encryption and decryption are carried out on-the-fly with minimal intervention from the processor. The encryption/decryption data are interchanged between the other Bluetooth low-level processing modules automatically.

Authentication Processing Accelerator

Processing-intensive authentication procedures are implemented in the hardware which reduce the time for authentication key production.

Receiver Signal Strength Indicator (RSSI)

This module provides a metric of the received signal strength of a Bluetooth transmission. It uses an 8-bit A/D to read the RSSI value of the transceiver.

64-byte TX and RX FIFOs

A 64-byte Transmit and a 64-byte Receive FIFO are used to buffer data between the Bluetooth baseband and the ARM processor.

Baseband TX and RX Processing Sequencer

Two dedicated sequencers are used, one for the Transmit direction and one for the Receive direction, these automate the Bluetooth packet transactions.

These sequencers, in conjunction with firmware running on the ARM core, execute the link control functions. The sequencer is used to free the ARM processor from the sequencing of bit-level processing. The processing needed by the various Bluetooth packets differs from packet type-to-packet type. The processing sequencers can be programmed according to the different packet types and carry out the required processing functions without further ARM processor intervention for the transmission of corresponding packet.

The baseband processor also eases the task of ensuring reliable data delivery. All bit-intensive tasks are done in hardware. The ARM core configures the modules and reads the results. Access code generation and correlation with received packet is carried out in hardware ensuring minimal decoding delay. With the firmware reading the results of a successful packet reception the HEC computation and header recognition is fully automated. The receive engine is automatically configured according to the packet header and, when required, payload header information, decodes the packet and stores it in the RX FIFO without further intervention from the processor.

8-bit ADC Interface for Bluetooth RX Data

An 8-bit external interface is used to connect an external analog-to-digital converter providing the digital representation of the analog signal decoded by the transceiver during reception. The analog signal is sampled at 4M samples per second requiring an external ADC capable of handling this sample rate.

Automatic DC-offset Cancellation and Symbol Recovery

Modern low-cost transceivers used in FSK applications, employing open loop modulation and/or demodulation together with limiter-discriminator detection at the receiver, produce output baseband signals which suffer from moderate to severe DC-offset fluctuation. The useful signal is a small fraction of the magnitude of the maximum DC-offset fluctuation. The number of symbols available to a Bluetooth baseband controller for DC-offset compensation and symbol timing recovery (STR) purposes are a mere 4 preamble bits, preceding the device access code (DAC) in every packet. Fast symbol timing acquisition is a prerequisite to successful decoding of the DAC and synchronization to the master device in a Bluetooth piconet. Due to this fact and in order to comply with tight slot timing requirements, a state-of-the-art DC-offset canceller, bit-symbol recovery circuit has been integrated. This module is capable of recovering the original Bluetooth bitstream in the noisiest of environments.

Voice CODEC

The voice CODEC module supports both CVSD (Continuous Variable Slope Delta) coding and log PCM coding (A-law and U-law). The coded voice data from both coding algorithms will be transferred with a constant bit rate of 64 kbits/sec.

The voice CODEC hardware consists of the following modules: A digital-to-analog converter (D/AC) for converting linear PCM data to the analog domain, an analog-to-digital converter (ADC), a digital interface for connection of external integrated voice codecs, a codec submodule, which implements the CVSD and log PCM alogrithms and two 32-byte FIFOs which are capable of holding 4 ms of coded voice data in each direction. Interrupts are generated when a programmable level of data in the FIFOs has been reached. Two externally connected low-pass filters are required to use a microphone and earphones.

Coded voice data from the Bluetooth interface can be transferred to an external voice codec through a digital interface. This interface can be configured as either a master or a slave. The interface accepts two possible clock formats: Short Frame Sync and Long Frame Sync.

USB Function Interface

The USB functionality is executed by an USB hardware block and firmware running on the ARM controller. This configuration allows acceleration of the intensive function processing while allowing flexibility in the implementation of higher level protocols over USB.

The USB hardware block consists of a Serial Interface Engine (SIE), a Serial Bus Controller (SBC) and a System Interface. 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 6 endpoints, each with dedicated double buffered FIFOs. One endpoint has an 8-byte FIFO, two endpoints have 16-byte FIFOs, two have 32-byte FIFOs and two have 128-byte FIFOs. The SBC manages the device address, monitors the status of the transactions, manages the FIFOs and communicates to the processor through a set of status and control registers. The System Interface connects the Serial Bus Controller to the processor.

16550 Compatible UART

The UART hardware module is a universal asynchronous receiver and transmitter with 16-byte Transmit and Receive FIFO. A programmable baud rate generator is provided to select, transmit and receive clock rates from 1200 bps to 921 Kbps.

The input clock to the baud rate generator is generated from a 96 MHz clock (derived from the internal clock generator). The required division ratios and the relative error are shown in Table 1 for a 96 MHz clock.

Table 1. Need Table Caption

Baud Rate	Divisor Used	% Error Between Desired and Actual
1200	5000	0
2400	2500	0
4800	1250	0
9600	625	0
19.2K	312	0.16
38.4K	156	0.16
57.6K	104	0.16
115.2K	52	0.16
230.4K	26	0.16
460.8K	13	0.16
921.6K	6.5	0.16

The UART module provides serial asynchronous receive data synchronization, parallel-to-serial and serial-to-parallel data conversions for both the transmitter and receiver sections. These functions are necessary for converting the serial data stream into parallel data which is required with digital data systems.

Synchronization for the serial data stream is accomplished by adding start and stop bits to the transmit data to form a data character (character orientated protocol). Data integrity is insured by attaching a parity bit to the data character for control and monitoring. The parity bit is checked by the receiver for any transmission bit errors.

Two 32-bit Timers

AT76C551 includes two identical system timers. Each system timer is a completely independent device. Each has adjustable prescale and preload capable of producing interrupts in a periodic or one-shot fashion.

Support for Different Operating Frequencies

The AT76C551 chip is able to operate with the following frequencies: 13 MHz, 14.4 MHz, 16.8 MHz and 19.44 MHz. Table 2 shows the correct configuration for each frequency.

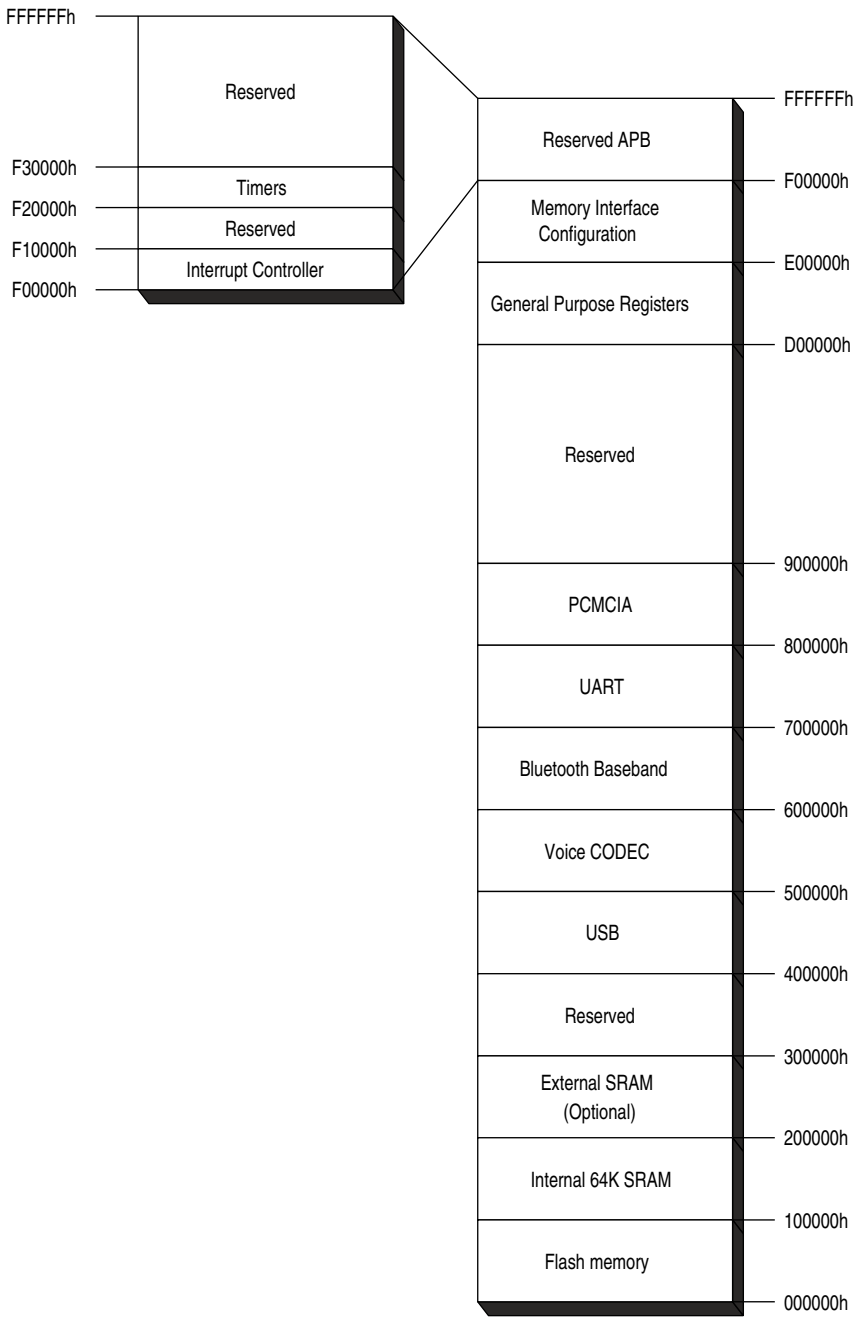
Either a crystal or an oscillator can be used. When the OSC_MODE pin is tied to ground the clock available at the EXT_OSC input is used to drive internal logic. When it is tied to VCC, the internal clock signal is derived from the XTAL1, XTAL2 pins and associated crystal and circuitry.

Also a very stable 13 MHz oscillator is required for transceiver operation. This is connected to the EXT_13_MHZ pin.

Table 2. Correct Configuration Frequencies

CLK_MODE0	CLK_MODE1	Crystal/Oscillator Frequency (MHz)
0	0	14.4
0	1	16.8
1	0	19.44
1	1	13

Figure 1. Memory Map of Prototype Version



Register Description

Memory Controller Register Set

The Memory Controller Register Set can be configured to cooperate with various types of external Flash and SRAM memories. The configuration parameters are held by Memory Configuration Registers (MCR) which are mapped into AMBA™ memory space. After AT76C551 resets, MCR default values guarantee correct operation of external Flash or SRAM memories connected to AT76C551. AT76C551 firmware adjusts MCR in order to achieve Flash and SRAM safe operation with minimum wait states.

In the reset configuration the external Flash can be found at the bottom of the address map. If the bit 0 of the remap register is set high, then the Memory Controller switches to the normal memory map in where there is internal SRAM.

SRAM Configuration Register (MCR0)

addr: E00000 hex R/W 8 bits

- **Bit 7 – Reserved**
- **Bits 6..4 – SWW[2:0]**

Number (0 - 7) of wait states during SRAM write cycles

- **Bit 3 – Reserved**
- **Bits 2..0 – SWR[2:0]**

Number (0 - 7) of wait states during SRAM read cycles

Note: Default Value: 77 hex

Flash Memory Configuration Register (MCR1)

addr: E00004 hex R/W 8 bits

- **Bits 7 – Reserved**
- **Bits 6..3**

Number (0 - 7) of wait states during write access of Flash memory latches

- **Bits 2..0 – FWR[2:0]**

Number (0 - 7) of wait states during Flash memory read cycles

Note: Default Value: 07 hex

Flash and Internal SRAM remap Register (MCR2)

addr: E00008 hex R/W 8 bits

- **Bits 7..1 – Reserved**
- **Bit 0 – REMAP**

Remap enable

Note: Default Value: 00 hex

PCMCIA Configuration Registers

The PCMCIA configuration registers are required by the PCMCIA standard. These registers are mapped into PCMCIA attribute memory space to allow the host to configure basic parameters of the PCMCIA device. They are accessible by the host but they are not accessible by ARM core.

COR: Configuration Option Register

PCMCIA addr: 0800 hex R/W 8 bits

- **Bit 7 – SRES: System Reset**

By setting this bit, the device is reset in a way equivalent to PCMCIA hardware reset signal activation.

Note: Tthis bit is not automatically cleared after set.

- **Bit 6 – IEVREQ**

Logic 1: Level mode interrupt.

Logic 0: Pulse mode interrupt

- **Bits 5..0 – CFX[5:0]: Configuration Index**

This field is written with the index number of the entry in the card's configuration table which the host selects. When all the field bits are zero, the device is in memory only mode.

Note: Default Value: 00 hex

CSR: Configuration and Status Register

PCMCIA addr: 0802 hex R/W 8 bits

- **Bits 7..6 – Reserved**
- **Bit 5 – IOIS8**

Logic 1: The host is only capable of 8-bit I/O accesses.

Logic 0: The host is capable of 8-bit and 16-bit I/O accesses.

- **Bits 4..0 – Reserved**

Note: Default Value: 00 hex

System Interface Registers

The System Interface Registers (SIR) lie in the PCMCIA interface unit. They are mapped into PCMCIA I/O space, i.e. they are directly accessible by the host but they are not directly accessible by the ARM core. They allow the host to configure and communicate with AT76C551 through host I/O space.

Note: All AMBA memory space (16M bytes address space) can be accessed by the host through the PCMCIA interface unit, via SIR1 - SIR5.

SIR0 – GCR: General Configuration Register

PCMCIA addr: 0000 hex R/W 8 bits

Bit 7 – SWRES: Software Reset

By setting this bit, all SIR registers are reset. However, AT76C551 units on the AMBA bus (ARM core, PAI, etc.) are not be reset by SWRES bit activation. This bit is automatically cleared after set.

- **Bit 6 – CORES: Core Reset**

While this bit is set, AT76C551 units on the AMBA bus (including ARM core) are held in reset state. When this bit is cleared, AT76C551 units on the AMBA bus exit reset state and ARM core, in particular, begins code execution by fetching its reset exception vector.

Note: This bit is not automatically cleared after it is set.

- **Bit 5 – 16/8-bit: 16/8-bit Access Mode**

If set, enables the 16-bit access of PCMCIA module with the system memory. If cleared, the access is 8-bit (see SIR1 - SIR5).

- **Bit 4 – Reserved**
- **Bit 3 – AIH: ARM Interrupt to Host**

This bit is set if an interrupt to the host has been generated by the ARM core (see bit AIH of MIR0) and is still pending. To acknowledge the interrupt and clear this bit, the host driver software must write 1 on this bit.

- **Bit 2 – HIA: HOst Interrupt to ARM**

When this bit is set by the host, an interrupt to the ARM core is generated (see bit HIA of MIR0). This bit is automatically cleared when the ARM core acknowledges the interrupt (see MIR0).

- **Bit 1 – AIHEN: ARM to Host Interrupt Enable**

Logic 0: Interrupts from ARM core to the host disabled

Logic 1: Interrupts from ARM core to the host enabled

- **Bit 0 – Reserved**

Note: Default Value: 00 hex – The host driver software is responsible for AT76C551 reset. The host driver software has to set CORES bit of GCR first. This will reset ARM core and all other units on AMBA bus. The host driver must next set SWRES bit of GCR. This will reset SIR and clear both CORES and SWRES bits, forcing ARM core to leave reset state and begin firmware program execution.

SIR1 – AMBA BSR: Bank Select Register

PCMCIA addr: 0001 hex R/W 8 bits

- **Bits 7..4 – AD[23:20]**

This register is used by the host in conjunction with SIR2, SIR3, SIR4 and SIR5 in order to access AMBA bus resources (Flash and internal or external SRAM) through the PCMCIA interface unit. Each time DLR (SIR4) is accessed, the AMBA Bank Select Register (BSR) drives AMBA address lines 23…20 and 18…15. AMBA address line 19 is always driven to logic 0.

- **Bits 3..0 – AD[18:15]**

Note: Default Value: 00 he

SIR2 – AMBA ALSR: Address Low Select Register

PCMCIA addr: 0002 hex R/W 8 bits

- **Bits 7..0 – AD[7:0]**

This register is used by the host in conjunction with SIR1, SIR3, SIR4 and SIR5 in order to access AMBA bus resources (Flash and internal or external SRAM) through the PCMCIA interface unit. Each time SIR4 is accessed, the AMBA ALSR drives AMBA address lines 7..0.

Note: Default Value: 00 hex

SIR3 – AMBA AHSR: Address High Select Register

PCMCIA addr: 0003 hex R/W 8 bits

- **Bit 7 – Reserved**
- **Bits 6..0 – AD[14:8]**

This register is used by the host in conjunction with SIR1, SIR2, SIR4 and SIR5 in order to access AMBA bus resources (Flash and internal or external SRAM) through the PCMCIA interface unit. Each time SIR4 is accessed, the AMBA AHSR drives AMBA address lines 14..8.

Note: Default Value: 00 he

SIR4 – AMBA I/O DLR: Data Low Register

PCMCIA addr: 0004 hex R/W 8 bits

- **Bits 7..0 – D[7:0]**

This register is used by the host in conjunction with SIR1, SIR2, SIR3 and SIR5 in order to access AMBA bus resources (Flash and internal or external SRAM) through the PCMCIA interface unit.

Note: Default Value: 00 hex

Writing the AMBA DLR initiates an AMBA write cycle. AMBA address lines are driven by SIR1, SIR2 and SIR3 while AMBA data lines are driven by SIR4 and SIR5. As a result an AMBA resource is written.

Reading the AMBA DLR initiates an AMBA read cycle. AMBA address lines are driven by SIR1, SIR2 and SIR3 while AMBA data lines are reflected to SIR4 and SIR5. As a result, an AMBA resource is read.

SIR5 – AMBA I/O DHR: Data High Register

PCMCIA addr: 0005 hex R/W 8 bits

- **Bits 7..0 – D[15:8]**

This register is used by the host in conjunction with SIR1, SIR2, SIR3 and SIR4 in order to access 16-bit AMBA bus resources (Flash and internal or external SRAM) through the PCMCIA interface unit.

Note: Default Value: 00 hex

Writing/reading SIR4 causes an AMBA write/read cycle. During this cycle the SIR4 is connected to AMBA data lines 7..0 while the AMBA Data High Register (DHR) is connected to AMBA data lines 15..8.

In order to write a 16-bit AMBA resource, the first most significant 8-bit write the AMBA DHR and second least significant 8-bit write the AMBA DLR. Writing to the AMBA DLR initiates the AMBA write cycle.

In order to read a 16-bit AMBA resource, the least stanching 8 bits must be read first for AMBA DLR and secondly the most significant 8 bits are read for AMBA DHR. Reading the AMBA DLR initiates the AMBA read cycle.

Note: Writing/reading only the AMBA DHR without writing/reading the AMBA DLR does not cause an AMBA write/read cycle.

- **SIR6 – GPR1: General Purpose Register 1 GPR1**

PCMCIA addr: 0006 hex W 8 bits

- **Bits 7..0 – GPR1[7:0]**

The host can only write GPR1. GPR1 bits 7..0 are reflected to bits 7..0 of MIR4 so theARM core can read them.

Note: Default Value: 00 hex

SIR7 – GPR2: General Purpose Register 2

PCMCIA addr: 0007 hex 8 bits

- **Bits 7..0 – GPR2[7:0]**

The host can only write GPR2. GPR2 bits 7..0 are reflected to bits 15..8 of MIR4 so the ARM core can read them.

Note: Default Value: 00 hex

SIR8 – GPR3: General Purpose Register 3

PCMCIA addr: 0008 hex W 8 bits

- **Bits 7..0 – GPR3[7:0]**

The host can only write GPR3. GPR3 bits 7..0 are reflected to bits 7..0 of MIR5 so the ARM core can read them.

Note: Default Value: 00 hex

In general, General Purpose Registers provide a means of one-way communication from the host driver software to AT76C551 firmware. Each GPR bit, or field, can carry any kind of information (configuration information, status information, command signaling) required by the interface between host driver software and AT76C551 firmware.

- **SIR9 – MR1: Mirror Register 1**

PCMCIA addr: 000E hex R 8 bits

- **Bits 7..0 – MR1[7:0]**

The host can only read MR1. MR1 bits 7..0 reflect bits 7..0 of MIR0, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR10 – MR2: Mirror Register 2

PCMCIA addr: 000F hex R 8 bits

- **Bits 7..0 – MR2[7:0]**

The host can only read MR2. MR2 bits 7..0 reflect bits 15..8 of MIR0, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR11 – MR3: Mirror Register 3

PCMCIA addr: 0010 hex R 8 bits

- **Bits 7..0 – MR3[7:0]**

The host can only read MR3. MR3 bits 7..0 reflect bits 7..0 of MIR1, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR12 – MR4: Mirror Register 4

PCMCIA addr: 0011 hex R 8 bits

- **Bits 7..0 – MR4[7:0]**

Note: Default Value: 00 hex – The host can only read MR4. MR4 bits 7..0 reflect bits 15..8 of MIR1, which can be written by the ARM core.

SIR13 – MR5: Mirror Register 5

PCMCIA addr: 0012 hex R 8 bits

- **Bits 7..0 – MR5[7:0]**

The host can only read MR5. MR5 bits 7..0 reflect bits 7..0 of MIR2, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR14 – MR6: Mirror Register 6

PCMCIA addr: 0013 hex R 8 bits

- **Bits 7..0 – MR6[7:0]**

The host can only read MR6. MR6 bits 7..0 reflect bits 15..8 of MIR2, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR15 – MR7: Mirror Register 7

PCMCIA addr: 0014 hex R 8 bits

- **Bits 7..0 – MR7[7:0]**

The host can only read MR7. MR7 bits 7..0 reflect bits 7..0 of MIR3, which can be written by the ARM core.

Note: Default Value: 00 hex

SIR16 – MR8: Mirror Register 8

PCMCIA addr: 0015 hex R 8 bits

- **Bits 7..0 – MR8[7:0]**

The host can only read MR8. MR8 bits 7..0 reflect bits 15..8 of MIR3, which can be written by the ARM core.

Note: Default Value: 00 hex

In general, Mirror Registers provide a means of one-way communication from AT76C551 firmware to the host driver software. See also MIR0 - MIR3.

MAC Interface Registers

MAC Interface Registers (MIR) lie in the PCMCIA interface unit. They are mapped into AMBA memory space, i.e. they are directly accessible by the ARM core but they are not directly accessible by the host. MIRs allow AT76C551 firmware to communicate with the host and to generate interrupts to the host processor.

MIR0 – PIR1: Processor Interface Register 1

addr: 800000 hex R/W 16 bits

- **Bits 15..4 – PIR1[15:4]**

General purpose I/O

Bits 15..8 are reflected to bits 7..0 of MR2 so that the host can read them.

Bits 7..0 are reflected to bits 7..0 of MR1 so that the host can read them.

- **Bit 3 – HIA: Host Interrupt to ARM**

This bit is set if an interrupt to the ARM core has been generated by the host (see bit HIA of SIR0) and is still pending. To acknowledge the interrupt and clear this bit, AT76C551 firmware must write 1 on this bit.

- **Bit 2 – AIH: ARM Interrupt to Host**

When this bit is set by the ARM core, an interrupt to the host is generated (see bit AIH of SIR0). This bit is automatically cleared when the host acknowledges the interrupt.

- **Bit 1 – HIAEN: Host to ARM Interrupt Enable**

Logic 0: Interrupts from the host to ARM core disabled

Logic 1: Interrupts from the host to ARM core enabled

- **Bit 0 – Reserved**

Note: Default Value: 0000 hex

MIR1 – PIR2: Processor Interface Register 2

addr: 800004 hex R/W 16 bits

- **Bits 15..8 – PIR2[15:8]**

General purpose I/O

MIR1 bits 15..8 are reflected to bits 7..0 of MR4 so that the host can read them.

- **Bits 7..0 – PIR2[7:0] General purpose I/O**

MIR1 bits 7..0 are reflected to bits 7..0 of MR3 so that the host can read them.

Note: Default Value: 0000 hex

Table MIR2 – PIR3: Processor Interface Register 3

addr: 800008 hex R/W 16 bits

- **Bits 15..8 – PIR3[15:8]**

General purpose I/O

MIR2 bits 15..8 are reflected to bits 7..0 of MR6 so that the host can read them.

- **Bits 7..0 – PIR3[7:0]**

General purpose I/O

MIR2 bits 7..0 are reflected to bits 7..0 of MR5 so that the host can read them.

Note: Default Value: 0000 hex

MIR3 – PIR4: Processor Interface Register 4

addr: 80000C hex R/W 16 bits

- **Bits 15..8 – PIR4[15:8]**

General purpose I/O

MIR3 bits 15..8 are reflected to bits 7..0 of MR8 so that the host can read them.

- **Bits 7..0 – PIR4[7:0]**

General purpose I/O

MIR3 bits 7..0 are reflected to bits 7..0 of MR7 so that the host can read them.

Note: Default Value: 0000 hex

MIR0 – MIR3 provide a means of one-way communication from AT76C551 firmware to the host driver software. Each MIR bit, or field, can carry any kind of information required by the interface between host driver software and AT76C551 firmware. The MIR4-MIR5 provide a means of one-way communication from the host driver software to AT76C551 firmware. See also GCR (SIR0) and GPR1 - GPR3 (SIR6 - SIR8).

MIR4 – PIR5: Processor Interface Register 5

addr: 800010 hex R/W 16 bits

- **Bits 15..8 – PIR5[15:8]**

General purpose I/O

MIR4 bits 15..8 reflect bits 7..0 of GPR2 (SIR7), which can be written by the host.

- **Bits 7..0 – PIR5[7:0]**

General purpose I/O

MIR4 bits 7..0 reflect bits 7..0 of GPR1 (SIR6), which can be written by the host.

Note: Default Value: 0000 hex

MIR5 – PIR6: Processor Interface Register 6

addr: 800014 hex R/W 16 bits

- **Bits 15..8 – PIR6[15:8]**

General purpose I/O

MIR5 bits 15..8 reflect bits 7..0 of GCR (SIR0), which can be written by the host.

- **Bits 7..0 – PIR6[7:0]**

General purpose I/O

MIR5 bits 7..0 reflect bits 7..0 of GPR3 (SIR8), which can be written by the host.

Note: Default Value: 0000 hex

Bluetooth Baseband Register Set

Bluetooth Baseband processor register file is mapped to the AMBA address space. Table 3 summarizes Bluetooth Baseband registers, grouped in functional sections.

Table 3. Bluetooth Register Set

Register	Addr. (hex)	Function
Packet Processing		
Address0	600000	Sets ADDR field used is various processes
Address1	600004	Access code, packet encoding, encryption, etc.
Pgrsp_counter	600008	Counter used in page response substate
Parity0	60000C	Set/Get the parity bits included in current access
Parity1	600010	Code/FHS packet
RSSI_CtrlStatus	600014	Control/Status of RSSI ADC
Controlstatus1	600018	Packet processing control/status 1 register
Controlstatus2	60001C	Packet processing control/status 2 register
Bluetooth Clock		
CLKN	600020	Native clock (free-running, RO)
CLOCK	600024	Provides estimated clock to frequency hopping
CLKPhase	600028	Native clock phase (free-running, RO)
CLKPhaseCorrelCorrect	60002C	Clock phase correction used at correlator trigger
CLKPhaseLimit	600030	Native clock phase limit (sets slot duration)
CLKPhaseWhenCorrel	600034	Native clock phase latched at correlator trigger
CLKControl	600038	Clock control register
CmpTimer_RxTxDataStart	60003C	RX/TX start compare timer
CmpTimer_GenPurpose	600040	General purpose compare timer
Transceiver Control		
RxTxSettleTimes	600044	Settle times for RX/TX power-up signals
TcCtrlStatus	600048	Transceiver control/status register
TcCommand	60004C	Transceiver command register
TcProgData	600050	Transceiver programming (via 3-wire bus) register
Bluetooth Interrupt		
IntMask	600054	Bluetooth interrupt mask register
IntStatus	600058	Bluetooth interrupt status register
IntClear	60005C	Bluetooth interrupt clear register
Payload FIFOs		
RxFifoCtrlStatus	600060	RX FIFO control/status register
RxFifoReadPort	600064	RX FIFO read port (8-bit)
TxFifoCtrlStatus	600068	TX FIFO control/status register
TxFifoWritePort	60006C	TX FIFO write port (8-bit)

Table 3. Bluetooth Register Set (Continued)

Register	Addr. (hex)	Function
Hop Frequency Selection		
HopSelCtrlStatus	600070	Hop selection control/status register
HopSelKern_ABCDE	600074	Hop selection kernel inputs A, B, C, D, E
HopSelKern_FXY1Y2	600078	Hop selection kernel inputs F, X, Y1, Y2
Rx Front		
RxFrontReg	60007C	Rx Front register
Packet Encryption		
Kc0	600080	Hold the encryption key used for data payload
Kc1	600084	Enciphering/deciphering
Kc2	600088	
Kc3	60008C	
E_functions		
ArReg0	600090	Sets configuration
ArReg1	600094	
ArReg2	600098	
ArReg3	60009C	
KeyReg0	6000A0	Set the half keys used in SAFER Ar rounds
KeyReg1	6000A4	
KeyReg2	6000A8	
KeyReg3	6000AC	
KeyReg4	6000B0	

Address0

addr: 600000 hex R/W 32 bits

- **Bits 31..0 – ADDR[31:0]**

Sets ADDR field, used in access code generation, packet encoding, encryption and frequency hopping.

Note: Default Value: 00000000 hex

Address1

addr: 600004 hex R/W 32 bits

- **Bits 15..0 – ADDR[47:32]**

Sets ADDR field, used in access code generation, packet encoding, encoding and frequency hopping.

- **Bits 31..16 – Reserved**

Note: Default Value: 00000000 hex

Pgrsp_counter

addr: 600008 hex R/W 32 bits

- **Bits 31..5 – Reserved**
- **Bits 4..0 – PageRspCounter[4:0]**

Sets page response counter value to be used for page response routine and data whitening initialization prior FHS packet transmission or reception.

Note: Default Value: 00000000 hex

Parity0

addr: 60000C hex R/W 32 bits

- **Bits 31..0 – ParityBits[31:0]**

When channel access code is generated from FHS syncword, sets the ParityBits[31:0] to be used in channel access code construction.

When channel access code is generated from BD_ADDR only, gets the ParityBits[31:0] of the channel access code.

Note: Default Value: 00000000 hex

Parity1

addr: 600010 hex R/W 32 bits

- **Bits 31..2 – Reserved**
- **Bits 1..0 – ParityBits[33:32]**

When channel access code is generated from FHS syncword, sets the ParityBits[33:32] to be used in channel access code construction.

When channel access code is generated from BD_ADDR only, gets the ParityBits[33:32] of the channel access code

Note: Default Value: 00000000 hex

RSSI_CtrlStatus

addr: 600014 hex R/W 32 bits

- **Bits 31..16 – Reserved**
- **Bits 15..8 R – ADC data**

Result of last conversion (last RSSI value sampled)

- **Bit 7 – R ADC Status**

Set by hardware when conversion procedure has been completed

- **Bits 6..4 – Reserved**
- **Bits 3..2 W – ADC Mode**

00: ADC idle

01: One shot conversion

10: Continuous conversion

- **Bit 1 – W ADC Start**

Set by firmware to start a conversion procedure when ADC mode is “one shot conversion” (auto-clear).

- **Bit 0 – W ADC Power**

0: ADC off

1: ADC on

Note: Default Value: 0000 hex

ControlStatus1

addr: 600018 hex R/W 32 bits

- **Bits 31..19 – Reserved**
- **Bits 18..9 – Packet_header**

Sets packet header before packet TX.

Gets packet header after packet header RX during packet RX.

- **Bits 8..0 – Payload_length**

Sets payload body length before data packet TX

Gets payload body length after payload header RX during packet RX

Note: Default Value: 00000000 hex

CtrlStatus2

addr: 60001C hex R/W 32 bits

- **Bits 31..20 – Reserved**
- **Bit 19 R – AccCodeBusy**

Set by hardware during access code calculation or transmission

- **Bit 18 R – CipherBusy**

Set by hardware during ciphering/deciphering

- **Bit 17 R – E_funBusy**

Set by hardware during E_function calculation

- **Bit 16 R – TxDataBusy**

Set by hardware while TX data is being streamed out

- **Bit 15 R – RxDataBusy**

Set by hardware while RX data is being streamed in

- **Bits 14...13 W – E_funType**

Selects the type of E function to be calculated

00: E1 (used for authentication)

01: E21 (used for link key generation)

10: E22 (used for link key generation)

11: E3 (used for encryption key generation)

- **Bit 12 R/W – EfunEnblBusy**

Set by firmware to start E function calculation

Reset by hardware when calculation is complete

- **Bit 11 W – CipherMode**

0: No encryption used

1: Packet payload is encrypted in TX and decrypted in RX

- **Bits 10..4 W – CorrelThresh**

Sliding correlator threshold value, default = 51

Sets sliding correlator sensitivity.

- **Bit 3 W– GetFHS_syncword**

Set by firmware to start access code recovery from LAP and FHS packet parity bits. Auto-clear

- **Bit 2 W – EvalAccCode**

Set by firmware to start access code calculation from BD_ADDR only. Auto-clear

- **Bit 1 W – AccCodeType**

0: Access code TXed (and loaded on sliding correlator) has no trailer bits, thus access code size is 68 bits.

1: Access code TXed (and loaded on sliding correlator) has trailer bits, thus access code size is 72 bits.

- **Bit 0 W – RxFHS**

0: The packet expected to be RXed is not an FHS packet.

1: An FHS packet RX is expected.

Note: Default Value: 00000000 hex

CLKN

addr: 600020 hex R 32 bits

Bits 31..28 – Reserved

- **Bits 27..0 – CLKN[27:0]**

Provides native clock current value to firmware.

Note: Default Value: 00000000 hex

CLOCK

addr: 600024 hex R/W 32 bits

- **Bits 31..28 – Reserved**
- **Bits 27..0 – CLOCK[27:0]**

In this register specific fields of device's native clock or of transmitter's estimated in receiver native clock are set.

Note: Default Value: 00000000 hex

CLKPhase

addr: 600028 hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bits 14..0 – CLKPhase[14:0]**

Provides native clock phase current value to firmware. Native clock phase is estimated in system clock cycles.

Note: Default Value: 00000000 hex

CLKPhaseCorrelCorrect

addr: 60002C hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bits 14..0 CLKPhase_, Correl_, Correct[14:0]**

Provides slave's hardware with the proper native clock phase value just after correlator trigger. At this moment slave's hardware automatically adjusts CLKN with CLKN of corresponding master.

Note: Default Value: 00000000 hex

CLKPhaseLimit

addr: 600030 hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bits 14..0 – CLKPhase_, Limit[14:0]**

Sets native clock phase maximum value in system clock cycles. Effectively sets half-slot duration in system clock cycles.

Note: Default Value: 00000000 hex

CLKPhaseWhenCorrel

addr: 600034 hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bits 14..0 – CLKPhase_, When_, Correl[14:0]**

Samples and holds native clock phase at correlator trigger. Thus enables slave-to-master clock drift estimation.

Note: Default Value: 00000000 hex

CLKCtrl

addr: 600038 hex R/W 32 bits

- **Bits 31..4 – Reserved**
- **Bit 3 – cmpCLKN0_invert**

0: Native clock bit 0 is not inverted for timer comparisons.

1: Native clock bit 0 is inverted for timer comparisons.

Bit 3 value is “don’t care” if bit 0 of the register is reset.

- **Bit 2 – ForcePhase_, Adjust**

Set by firmware to force native clock phase adjustment, i.e. set native clock phase equal to CLKPhaseCorrelCorrect register contents. Auto-clear.

- **Bit 1 – AutoPhaseAdjust**

0: Native clock phase is not auto-adjusted.

1: Native clock phase is auto-adjusted, i.e. set equal to CLKPhaseCorrelCorrect register contents just after each correlator trigger.

- **Bit 0 – cmpCLKN0_ enable**

0: Compare timer event is generated when compare timer bits 14 - 0 are equal to native clock phase.

1: Compare timer event is generated when compare timer bits 14 - 0 are equal to native clock phase AND compare timer bit 15 is equal to native clock bit 0 (possibly inverted).

Note: Default Value: 00000000 hex

CmpTimer_RxTxStart

addr: 60003C hex R/W 32 bits

- **Bits 15..0 – CmpTimer_, RxTxStart[15:0]**

Sets compare timer for packet RX or TX procedure start

Note: Default Value: 00000000 hex

CmpTimer_GenPurpose

addr: 600040 hex R/W 32 bits

- **Bits 15..0 – CmpTimer_, GenPurpose[15:0]**

Sets general purpose compare timer

Note: Default Value: 00000000 hex

Compare timers are used to notify an event during a full-slot or a half-slot. Each compare timer is compared to current native clock phase. Optionally, MSB of each compare timer can be compared to bit 0 of native clock (possibly inverted).

When compare timer matches current native clock phase and current native clock bit 0, the corresponding event is generated.

CmpTimer_RxTxStart generates a maskable interrupt and starts RX or TX procedure, if the corresponding bit of TcCommand register has been set.

CmpTimer_GenPurpose only generates a maskable interrupt.

RxTxSettleTimes

addr: 600044 hex R/W 32 bits

- **Bits 31..24 – Reserved**
- **Bits 23..16 –** TC_TXON**, settleTime[7:0]**

Microseconds from TC_TXON rise to outgoing packet data transmission.

- **Bits 15..8 –** TC_RXON**, settleTime[7:0]**

Microseconds from TC_RXON rise to correlator activation.

- **Bits 7..0 –** TC_PURX/TX**, settleTime[7:0]**

Microseconds from TC_PURX/TX rise to TC_TXON or TC_RXON rise.

Note: Default Value: 00000000 hex

TcCtrlStatus

addr: 600048 hex R/W 32 bits

- **Bits 31..9 – Reserved**
- **Bit 8** TC_LD

Reflects transceiver TC_LD pin level

- **Bit 7 R/W –** TC_PUREG

Sets transceiver TC_PUREG pin level

- **Bit 6 R/W –** TC_PUVCO

Sets transceiver TC_PUVCO pin level

- **Bit 5 R/W – TC_PUPLL**

Sets transceiver TC_PUPLL pin level

- **Bit 4 R/W – TC_PURXTX**

Sets transceiver TC_PURXTX pin level, when bit 1 of this register is reset

- **Bit 3 R/W –** TC_RXON

Sets transceiver TC_RXON pin level, when bit 1 of this register is reset

- **Bit 2 R/W –** TC_TXON

Sets transceiver TC_TXON pin level, when bit 1 of this register is reset

- **Bit 1 R/W – RX/TX_auto**

Selects control mode for transceiver signals TC_RXON, TC_TXON & TC_PURXTX

1: The transceiver signals are controlled automatically.

0: The transceiver signals are set to reflect bits 4...2 of this register.

- **Bit 0 R/W –** TC_I_CP_SW

Sets transceiver TC_I_CP_SW pin level

Note: Default Value: 00000000 hex

TcCommand

addr: 60004C hex W 32 bits

- **Bits 31..3 – Reserved**
- **Bit 2 – RxEnable**

Set by firmware to start a RX cycle at then next event generated by compare timer CmpTimer_RxTxStart

- **Bit 1 – TxEnable**

Set by firmware to start a RX cycle at then next event generated by compare timer CmpTimer_RxTxStart.

- **Bit 0 – RxTx_, Abort**

Set by firmware to immediately abort current RX or TX cycle, or to cancel the programmed RX or TX cycle due to start.

Note: Default Value: 00000000 hex

Provided that bit 1 of TcCtrlStatus register is set the Bluetooth baseband processor has the ability to automatically generate the sequence of signals necessary to implement TX and RX.

To initiate an automatic RX cycle, the firmware has to program CmpTimer_RxTxStart register and then set bit 2 of TcCommand register. When the compare timer generates its event the RX cycle begins and PU_RX/TX signal is raised. After PU_RX/TX settle time the RX_ON signal is raised. After RX_ON settle time baseband data acquisition circuits are enabled and the correlator begins to search for a known access code. If correlator is triggered, then the correlator is powered down and RX packet data is acquired. After RX packet completion the RX cycle ends and the transceiver is powered down.

Note: The transceiver and the correlator will be held active until correlator trigger occurs, or RX cycle is aborted by firmware, i.e. firmware sets bit 0 of TcCommand register.

To initiate an automatic TX cycle the firmware has to program CmpTimer_RxTxStart register and then set bit 1 of the TcCommand register. When the compare timer generates its event, the TX cycle begins and PU_RX/TX signal is raised. After PU_RX/TX settle time the TX_ON signal is raised causing TX ramp rise. While TX ramp rises, the transmitter is fed with a 0/1-alternating data stream. After TX_ON settle time (which should include TX ramp set time), the actual TX packet data stream is given to the transceiver. After TX packet completion, the TX cycle ends and the transceiver is powered down.

A RX or TX automatic cycle can be aborted at any moment. To abort a current RX or TX cycle the firmware will set bit 0 of TcCommand register. This will power down the transceiver and reset all baseband internal state machines to an idle state. To cancel the program, which has not yet started, RX or TX automatic cycle firmware can also use bit 0 of TcCommand register.

Note: Please note that the baseband is aware of RX or TX packet type and length therefore it has the ability to trace RX or TX packet fields and decide the FEC encoding, CRC calculation, etc.

TcProg

addr: 600050 hex R/W 32 bits

- **Bits 31..26 – Reserved**
- **Bit 25 – 3wb_, EnableBusy**

Set by firmware to initiate serial data send on the 3-wire bus. Reset by hardware to indicate that 3-wire bus access is complete.

- **Bit 2 – 3wb_,ClockRate**

0: 3-wire bus clock is generated by system clock division by 4.

1: 3-wire bus clock is generated by system clock division by 8.

- **Bits 23..0 – 3wb_, Data[23:0]**

24-bit data word to be send on the 3-wire bus. MSB is sent first.

Note: Default Value: 00000000 hex

IntStatus

addr: 600054 hex R 32 bits

- **Bits 31..13 – Reserved**
- **Bit 12 – TxFifoAlmEmpty_IntStatus**

TX FIFO almost empty interrupt status.

- **Bit 11 – RxFifoAlmFull_IntStatus**

RX FIFO almost full interrupt.

- **Bit 10 – GenPurpTim_ IntStatus**

General purpose compare timer interrupt.

- **Bit 9 – TxAcCodeComplete_IntStatus**

RX/TX start compare timer interrupt enable.

- **Bit 8 – RxTxStart_ IntStatus**

RX/TX start compare timer interrupt.

- **Bit 7 – TxPktComplete_IntStatus**

TX packet completion interrupt.

- **Bit 6 – RxCorrelTrig_ IntStatus**

Correlator trigger interrupt.

- **Bit 5 – RxHecFail_IntStatus**

HEC fail interrupt.

- **Bit 4 – RxCrcFail_ IntStatus**

CRC fail interrupt.

- **Bit 3 – RxFecFail_ IntStatus**

FEC fail interrupt.

- **Bit 2 – RxPktHeaderRdy_IntStatus**

RX packet header arrival interrupt.

- **Bit 1 – RxPayHeaderRdy_IntStatus**

RX payload header arrival interrupt.

- **Bit 0 – RxPayloadRdy_IntStatus**

RX payload completion interrupt.

Note: Default Value: 00000000 hex

IntMask

addr: 600058 hex R/W 32 bits

- **Bits 31..13 – Reserved**
- **Bit 12 – TxFifoAlmEmpty_IntEnable**

TX FIFO almost empty interrupt enable.

- **Bit 11 – RxFifoAlmFull_ IntEnable**

RX FIFO almost full interrupt enable.

- **Bit 10 – GenPurpTim_IntEnable**

General purpose compare timer interrupt enable.

- **Bit 9 – TxAcCodeComplete_IntEnable**

TX of Access Code has been completed.

- **Bit 8 – RxTxStart_ IntEnable**

RX/TX start compare timer interrupt enable.

- **Bit 7 – TxPktComplete_IntEnable**

TX packet completion interrupt enable.

- **Bit 6 – RxCorrelTrig_IntEnable**

Correlator trigger interrupt enable.

- **Bit 5 – RxHecFail_IntEnable**

HEC fail interrupt enable.

- **Bit 4 – RxCrcFail_IntEnable**

CRC fail interrupt enable.

- **Bit 3 – RxFecFail_IntEnable**

FEC fail interrupt enable.

- **Bit 2 – RxPktHeaderRdy_IntEnable**

RX packet header arrival interrupt enable.

- **Bit 1 – RxPayHeaderRdy_IntEnable**

RX payload header arrival interrupt enable.

- **Bit 0 – RxPayloadRdy_IntEnable**

RX payload completion interrupt enable.

Note: Default Value: 00000000 hex

IntClear

addr: 60005C hex W 32 bits

- **Bits 31..11 – Reserved**
- **Bit 10 – GenPurpTim_IntClear**

Clears general purpose compare timer interrupt

- **Bit 9 – TxAcCodeComplete_IntClear**

Clears access Code Tx completed interrupt

- **Bit 8 – RxTxStart_IntClear**

Clears RX/TX start compare timer interrupt

- **Bit 7 – TxPktComplete_IntClear**

Clears TX packet completion interrupt

- **Bit 6 – RxCorrelTrig_IntClear**

Clears correlator trigger interrupt

- **Bit 5 – RxHecFail_IntClear**

Clears HEC fail interrupt

- **Bit 4 – RxCrcFail_IntClear**

Clears CRC fail interrupt

- **Bit 3 – RxFecFail_IntClear**

Clears FEC fail interrupt

- **Bit 2 – RxPktHeaderRdy_IntClear**

Clears RX packet header arrival interrupt

- **Bit 1 – RxPayHeaderRdy_IntClear**

Clears RX payload header arrival interrupt

- **Bit 0 – RxPayloadRdy_IntClear**

Clears RX payload completion interrupt

All events related to packet RX/TX as well as events generated by compare timers can activate the “Bluetooth Baseband” system interrupt. Conditions related to Bluetooth Baseband FIFOs can activate “Bluetooth Baseband FIFOs” system interrupt.

Each event related to packet RX/TX, or generated by a compare timer, is always latched on the corresponding bit of IntStatus register. The bit is set when the event occurs. The bit is reset when the corresponding bit of IntClear register is set by the firmware to acknowledge the event.

However, an event related to packet RX/TX or generated by a compare timer will produce an interrupt only if the corresponding bit of IntMask register has been set by firmware.

Conditions related to Bluetooth Baseband FIFOs are “TX FIFO almost full” and “RX FIFO almost empty”. These conditions are not latched, but just reflected on the corresponding bits of IntStatus register. Bluetooth Baseband FIFO conditions have no corresponding bits in IntClear register, because they will be automatically reset after proper FIFO service by firmware.

Bluetooth Baseband FIFO conditions will produce an interrupt only if the corresponding bit of IntMask register has been set by firmware.

RxFifoCtrlStatus

addr: 600060 hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bit 14 R – Empty**

Set by hardware while FIFO is empty

- **Bit 13 R Full – Set by hardware while FIFO is full**
- **Bit 12 W – Reset**

Set by firmware to discard any possible FIFO contents. Auto-clear.

- **Bits 11..6 R/W – Threshold[5...0]**

Set by firmware to define the minimum level at which FIFO is considered “almost full”

- **Bits 5..0 R – Level[5...0]**

Current FIFO level, i.e. number of received bytes into FIFO waiting to be read

Note: Default Value: 00000000 hex

RxFifoReadPort

addr: 600064 hex R 32 bits

- **Bits 31..8 – Reserved**
- **Bits 7..0 – RX_byte[7:0]**

Least recent received byte not yet read

Note: Please note that reading RxFifoReadPort register is meaningful only if RX FIFO level is at least one, i.e. at least one byte exists in RX FIFO.

If RX FIFO is not serviced before it is full it will overflow and subsequent bytes received will be lost. RX FIFO capacity is 64 bytes.

TxFifoCtrlStatus

addr: 600068 hex R/W 32 bits

- **Bits 31..15 – Reserved**
- **Bit 14 R – Empty**

Set by hardware while FIFO is empty

- **Bit 13 R – Full**

Set by hardware while FIFO is full

- **Bit 12 W – Reset**

Set by firmware to discard any possible FIFO contents. Auto-clear

- **Bits 11..6 R/W – Threshold[5...0]**

Set by firmware to define the maximum level at which FIFO is considered “almost empty”.

- **Bits 5..0 R – Level[5...0]**

Current FIFO level, i.e. number of bytes written into FIFO, waiting to be transmitted.

Note: Default Value: 00000000 hex

TxFifoWritePort

addr: 60006C hex W 32 bits

- **Bits 31..8 – Reserved**
- **Bits 7..0 – TX_byte[7:0]**

Next byte to transmit

Note: Writing TxFifoWritePort register is not permitted if TX FIFO is full.

If TX FIFO is not serviced before it is empty it will underflow and transmitted bytes will be lost. TX FIFO capacity is 64 bytes.

hopSelCtrlStatus

addr: 600070 hex RW 32 bits

Bits 31..4 – Reserved

- **Bit 4..3 – Hop_Type**

Indicates frequency hop type (page, inquiry, etc.).

- **Bit 2 – Hop_Kernel_Direct**

0: Hop selection calculation parameters (A,B,.X,Y1,.) are estimated by hardware.

1: Hop selection calculation parameters are provided directly by firmware.

- **Bit 1 – Mode_79_23**

0: 79 Frequency Hop System

1: 23 Frequency Hop System

- **Bit0 – HopSelEnable_Busy**

Set by firmware and enables hardware frequency pointer calculation. Cleared by hardware when calculation is finished

Note: Default Value: 00000000 hex

hopSel_ABCDE

addr: 600074 hex RW 32 bits

- **Bits 31..30 – Reserved**
- **Bits 29..25 – A[4:0]**

Hop kernel register A

- **Bits 24..21 – B[4:0]**

Hop kernel register B

- **Bits 20..16 – C[4:0]**

Hop kernel register C

- **Bits 25..7 – D[4:0]**

Hop kernel register D

- **Bits 6..0 – E[4:0]**

Hop kernel register E

Note: Default Value: 00000000 hex

hopSel_FXY1Y2

addr: 600078 hex RW 32 bits

- **Bits 31..30 – Reserved**
- **Bits 29..25 – F[4:0]**

Hop kernel register F

- **Bits 24..21 – X4:0]**

Hop kernel register X

- **Bits 20..16 – Y14:0]**

Hop kernel register Y1

- **Bits 25..7 – Y24:0]**

Hop kernel register Y2

Note: Default Value: 00000000 hex

RxFrontReg

addr: 60007C hex W 32 bits

Bits 31..24 – Reserved

- **Bits 22..20 – RxFront_n_C[3:0]**
- **Bits 18..16 – RxFront_n_A[3:0]**
- **Bits 15..8 – RxFrontThresh_A[7:0]**
- **Bits 7..0 – RxFrontThreshB[7:0]**

Note: Default Value: 00000000 hex

Kc0

addr: 600080 hex W 32 bits

- **Bits 31..0 – Kc0[31:0]**

Key used for encryption

Note: Default Value: 00000000 hex

Kc1

addr: 600084 hex W 32 bits

- **Bits 31..0 – Kc1[31:0]**

Key used for encryption

Note: Default Value: 00000000 hex

Kc2

addr: 600088 hex W 32 bits

- **Bits 31..0 – Kc2[31:0]**

Key used for encryption

Note: Default Value: 00000000 hex

Kc3

addr: 60008C hex W 32 bits

- **Bits 31..0 – Kc3[31:0]**

Key used for encryption

Note: Default Value: 00000000 hex

ArReg0

addr: 600090 hex RW 32 bits

- **Bits 31..0 – ArReg0[31:0]**

Used in E_functions.Initialized by firmware and is set by hardware with E_function result.

Note: Default Value: 00000000 hex

ArReg1

addr: 600094 hex RW 32 bits

- **Bits 31..0 – ArReg1[31:0]**

Used in E_functions. Initialized by firmware and is set by hardware with E_function result.

Note: Default Value: 00000000 hex

ArReg2

addr: 600098 hex RW 32 bits

- **Bits 31..0 – ArReg2[31:0]**

Used in E_functions. Initialized by firmware and is set by hardware with E_function result.

Note: Default Value: 00000000 hex

ArReg3

addr: 60009C hex RW 32 bits

- **Bits 31..0 – ArReg3[31:0]**

Used in E_functions. Initialized by firmware and is set by hardware with E_function result.

Note: Default Value: 00000000 hex

KeyReg0

addr: 60000 hex RW 32 bits

- **Bits 31..0 – KeyReg0[31:0]**

Key used in E_functions

Note: Default Value: 00000000 hex

KeyReg1

addr: 6000A4 hex RW 32 bits

- **Bits 31..0 – KeyReg1[31:0]**

Key used in E_functions

Note: Default Value: 00000000 hex

KeyReg2

addr: 6000A8 hex RW 32 bits

- **Bits 31..0 – KeyReg2[31:0]**

Key used in E_functions

Note: Default Value: 00000000 hex

KeyReg3

addr: 6000AC hex RW 32 bits

- **Bits 31..0 – KeyReg3[31:0]**

Key used in E_functions

Note: Default Value: 00000000 hex

KeyReg4

addr: 6000B0 hex RW 32 bits

Voice CODEC Register Set

The Voice CODEC has four modes: CVSD, A-law, U-law and pass through where data is transferred out through the digital interface. The module has independent receive and transmit paths – each having a dedicated 32-byte FIFO.

VC_CTRL: Voice CODEC Analog Control

addr: 500000 hex R/W 13 bits

- **Bits 15..13 – Reserved**
- **Bit 12 – Rate**

1= 64 kHz sampling rate

0= 8 kHz sampling rate

- **Bits 11..10 – ADC_TC[1:0]**

Type of AD conversion:

00 = Reserved

01 = Reserved

10 = Reserved

11 = Conversion initiated by 64/8 kHz clock

- **Bit 9 – ADC_SC**

Start conversion

- **Bit 8 – ADC_EN: Enables ADC Converter**
- **Bit 7 – DAC_EN: Enables DAC Converter**
- **Bit 6 – DIG_INT: Enables Digital Interface**
- **Bit 5 – AL_DEC: Enables A-law Decoder**
- **Bit 4 – UL_DEC: Enables U-law Decoder**
- **Bit 3 – CV_DEC: Enables CVSD Decoder**
- **Bit 2 – AL_ENC: Enables A-law Encoder**
- **Bit 1 – UL_ENC: Enables U-law Encoder**
- **Bit 0 – CV_ENC: Enables CVSD encoder**

Note: Default Value: 0000 hex

Before conversions can take place the ADC and the DAC must be powered up. The ADC requires 10 µs power-up time while the DAC 20 µs power-up time. During this time, the coding modules must not be enabled. After powering up, the ADC/DAC conversion rate and the required coding mode can be programmed.

VCDI_MODE: Voice CODEC Digital Interface Control

addr: 500004 hex R/W 16 bits

Bits 15..6 – Reserved

- **Bit 5 – ENA: Enables Serial Transmission**

1= Enabled

0= Not enabled

- **Bit 4 – OSF: Output Sample Format**

1 = 16 bits

0 = 8 bits

- **Bit 3 – DIR: Audio Clock Direction**

1= Digital interface is slave

0= Digital interface is master

- **Bit 2 – BDIV: Divides Audio Clock**

Divides Audio Clock (depending on mode[4] bit) in order to drive the bit clock.

{mode[2], mode[4]} = 00, abclk = aclk/32

{mode[2], mode[4]} = 01, abclk = aclk/16

{mode[2], mode[4]} = 10, abclk = aclk/24

{mode[2], mode[4]} = 11, abclk = aclk/12

- **Bits 1..0 – M_CDC[1:0]: Master Audio Clock Divider Control**

00 = aclk is driven by mclock

01 = aclk = mclock/2

10 = aclk = mclock/4

11 = aclk = mclock/8

Note: Default Value: 0000 hex

The digital interface is able to operate either as a master or a slave. When operating as a master, a 6.144 MHz clock is used, while in slave mode, the clock available at the ABCLK_IN pin is used to drive the digital interface.

VC_INT_CTRL_STATUS: Voice CODEC Interrupt Control Status

addr: 500008 hex R/W 16 bits

- **Bits 15..4 – Reserved**
- **Bit 3 – EN_RC_I:Enable Voice CODEC Transmit FIFO Interrupt**
- **Bit 2 – EN_TR_I: Enable Voice CODEC Receive FIFO Interrupt**
- **Bit 1 – VCRC_S: Transmit FIFO Interrupt Status**
- **Bit 0 – VCTR_S: Receive FIFO Interrupt Status**

Note: Default Value: 0000 hex

VC_RxFifoCtrl: Voice CODEC Receive FIFO Control Status Register ()

addr: 50000C hex R/W 16 bits

- **Bit 15 – EMPTY**
- **Bit 14 – ALEMPTY**

1 byte left to be transmitted.

- **Bit 13 – FULL**
- **Bit 12 – RESFF: Reset Receive FIFO**
- **Bits 11..10 – Reserved**
- **Bits 8..5 – RCFTR[4:0]: Receive FIFO Threshold**
- **Bits 1..0 – LEVEL[4:0]: Receive FIFO Level**

Note: Default Value: 0000 hex

VC_RxReadPort: Voice CODEC Receive FIFO Read Port

addr: 500010 hex R 8 bits

- **Bits 7..0 – VC_RXD[7:0]: Receive Data**

Note: Default Value: 00 hex

VC_TxFifoCtrl: Voice CODEC Transmit FIFO Control Status Register

addr: 500014 hex R/W 16 bits

- **Bit 15 – EMPTY**
- **Bit 14 – ALEMPTY**

1 byte left to be transmitted.

- **Bit 13 – FULL**
- **Bit 12 – RESFF: Reset Transmit FIFO**
- **Bits 11..10 – Reserved**
- **Bits 9..5 – TXFTR[4:0]: Transmit FIFO threshold**
- **Bits 4..0 – LEVEL[4:0]: Transmit FIFO Level**

Note: Default Value: 0000 hex

VC_TxWritePort: Voice CODEC Transmit FIFO Write Port

addr: 500018 hex W 8 bits

- **Bits 7..0 – VC_RXD[7:0]: Transmit Data**

Note: Default Value: 00 hex

USB Registers – USB Wrapper Registers

The following registers are found in the USB wrapper block and control the overall performance of the USB hardware block. They provide status information, allow interrupt masking and DMA programming for fast data transfers between the DPRAM and the endpoint buffers.l

Table 4. Summary of the USB Cell Specific Registers

Register	Address	Default	Function
SLP_MD_EN	5000000h	00000000	Sleep mode control
GLB_IRQ_MSK	5000004h	00000000	Global Interrupt Master register
IRQ_STAT	5000008h	00000000	Master Interrupt Status
RES_STAT	500000Ch	00000000	Reset Status
DEF_EP_PAIRS	5000010h	00000000	Endpoint Pairs Definition
USB_RDMA_LEN	500001Ch	00000000	Receive DMA packet length requested
USB_DMA_FADD	5000020h	00000000	DMA target Endpoint address
USB_RDMA_LENR	5000024h	00000000	Receive DMA packet length transferred
USB_RDMA_EN	5000028h	00000000	Receive DMA Enable
USB_DMA_RADD	5000040h	00000000	RAM target address for DMA cycles
USB_TDMA_LEN	5000044h	00000000	Transmit DMA packet length requested
USB_TDMA_LENR	5000048h	00000000	Transmit DMA packet length transferred
USB_TDMA_EN	500004Ch	00000000	Transmit DMA Enable
FRM_NUM_H	50003F4h	xxxxx000	Frame Number High Register
FRM_NUM_L	50003F0h	xxxxx000	Frame Number Low Register
GLB_STATE	50003ECh	xxxxx000	Global State Register
SPRSR	50003E8h	xxxxx000	Suspend/Resume Register
SPRSIE	50003E4h	xxxxxx00	Suspend/Resume Interrupt Enable Register
UISR	50003DCh	00000000	USB Interrupt Status Register
UIAR	50003D4h	xxxxx000	USB Interrupt Acknowledge Register
UIER	50003CCh	xxxxx000	USB Interrupt Enable Register
FADDR	50003C8h	00000000	Function Address Register
ENDPPGPG	50003C4h	00000000	Function Endpoint Ping-pong Register
ECR0	50003BCh	0xxx0000	Endpoint0 Control Register
ECR1	50003B8h	0xxx0000	Endpoint1 Control Register
ECR2	50003B4h	0xxx0000	Endpoint2 Control Register
ECR3	50003B0h	0xxx0000	Endpoint3 Control Register
ECR4	50003ACh	0xxx0000	Endpoint4 Control Register
ECR5	50003A8h	0xxx0000	Endpoint5 Control Register
ECR6	50003A4h	0xxx0000	Endpoint6 Control Register
CSR0	500037Ch	x1110000	Endpoint0 Control and Status Register
CSR1	5000378h	x1110000	Endpoint1 Control and Status Register

Table 4. Summary of the USB Cell Specific Registers (Continued)

Register	Address	Default	Function
CSR2	5000374h	x1110000	Endpoint2 Control and Status Register
CSR3	5000370h	x1110000	Endpoint3 Control and Status Register
CSR4	500036Ch	x1110000	Endpoint4 Control and Status Register
CSR5	5000368h	x1110000	Endpoint5 Control and Status Register
CSR6	5000364h	x1110000	Endpoint6 Control and Status Register
FDR0	500033Ch	00000000	FIFO 0 Data Register
FDR1	5000338h	00000000	FIFO 1 Data Register
FDR2	5000334h	00000000	FIFO 2 Data Register
FDR3	5000330h	00000000	FIFO 3 Data Register
FDR4	500032Ch	00000000	FIFO 4 Data Register
FDR5	5000328h	00000000	FIFO 5 Data Register
FDR6	5000324h	00000000	FIFO 6 Data Register
FBYTE_CNT0_L	50002FCh	00000000	Byte Count FIFO 0 Register [7:0]
FBYTE_CNT1_L	50002F8h	00000000	Byte Count FIFO 1 Register [7:0]
FBYTE_CNT2_L	50002F4h	00000000	Byte Count FIFO 2 Register [7:0]
FBYTE_CNT3_L	50002F0h	00000000	Byte Count FIFO 3 Register [7:0]
FBYTE_CNT4_L	50002ECh	00000000	Byte Count FIFO 4 Register [7:0]
FBYTE_CNT5_L	50002E8h	00000000	Byte Count FIFO 5 Register [7:0]
FBYTE_CNT6_L	50002E4h	00000000	Byte Count FIFO 6 Register [7:0]
FBYTE_CNT0_H	50002BCh	00000000	Byte Count FIFO 0 Register [10:8]
FBYTE_CNT1_H	50002B8h	00000000	Byte Count FIFO 1 Register [10:8]
FBYTE_CNT2_H	50002B4h	00000000	Byte Count FIFO 2 Register [10:8]
FBYTE_CNT3_H	50002B0h	00000000	Byte Count FIFO 3 Register [10:8]
FBYTE_CNT4_H	50002ACh	00000000	Byte Count FIFO 4 Register [10:8]
FBYTE_CNT5_H	50002A8h	00000000	Byte Count FIFO 5 Register [10:8]
FBYTE_CNT6_H	50002A4h	00000000	Byte Count FIFO 6 Register [10:8]

SLP_MD_EN: Sleep Mode Control

addr 5000000h R/W 8 bits

- **Bits 7..6 – Reserved**
- **Bit 5 – SLP**

Put the USB module in sleep mode

- **Bits 4..0 – Reserved**

Note: Default: 00h

GLB_IRQ_MSK: Global Interrupt Master Register

addr 5000004h R/W 16 bits

- **Bits 15..11 – Reserved**
- **Bit 10 – TDMA_IEN:Transmit DMA Interrupt Enable**
- **Bit 9 – RDMA_IEN: Receive DMA Interrupt Enable**
- **Bit 8 – URES_INT**

When this bit is high, the USB reset interrupts are enabled

- **Bit 7 – Reserved**
- **Bit 6 – INT_EN**

When this bit is high, it allows the INTERRUPT line from the USB protocol handler to cause interrupts

- **Bits 5..2 – Reserved**
- **Bit 1 – SUSP_INT**

If this bit is high, an interrupt is generated when the USB enters suspend mode.

- **Bit 0 – RSM_INT**

If this bit is high, an interrupt is generated when the USB enters resume mode.

Note: Default: 00h

These registers allow interrupt masking for the following interrupt sources:

1. INTERRUPT line from USB protocol handler.
2. USB reset (USB_RES). A USB reset signal is asserted from USB protocol handler when the USB host requests it by forcing both the differential USB network signals to low level.
3. Suspend: A USB device enters in suspend only when requested by the USB host through bus inactivity for at least 3 ms.
4. Resume: a J to K state change on the USB port signals resume.

IRQ_STAT: Master Interrupt Status

addr 5000008h R 16 bits

- **Bits 15..11 – Reserved**
- **Bit 10 – TDMA_TC: Transmit DMA Complete**
- **Bit 9 – RDMA_TC: Receive DMA Complete**
- **Bit 8 – SUS_RES_ST**

Indicates the current status of the USB block. When this bit is high, the USB is in Suspend mode while, when low, the USB has resumed.

- **Bit 7 – Reserved**

- **Bit 6 – INTER_LINE**

The INTERRUPT line from the USB protocol handler is asserted.

- **Bits 5..2 – Reserved**
- **Bit 1 – SUSP**

When this bit is high, the USB has entered the suspend state.

- **Bit 0 – RSM**

When this bit is high, the USB has entered the resume state.

Note: Default: 00h

RES_STAT: Reset Status

addr 500000Ch R/W 8 bits

- **Bits 7..5 – Reserved**
- **Bit 4 – USB_RES**

Set when USB module enters reset state.

- **Bits 3..0 – Reserved**

Note: Default: 00h

DEF_EP_PAIRS: Endpoint Pairs Definition

addr 5000010h R/W 8 bits

- **Bits 7...4 – Reserved**
- **Bit 3 – EP3_EN_PAIR**

When this bit is high, the EP3 supports an OUT and an IN endpoint. In this case the EP6 is used as the IN endpoint.

- **Bit 2 – EP2_EN_PAIR**

When this bit is high, the EP2 supports an OUT and an IN endpoint. In this case the EP5 is used as the IN endpoint.

- **Bit 1 – EP1_EN_PAIR**

When this bit is high, the EP1 supports an OUT and an IN endpoint. In this case the EP4 is used as the IN endpoint

- **Bit 0 – Reserved**

Note: Default: 00h

This 3-bit register defines which endpoints support both IN and OUT connections. Only the endpoints 1 - 3 support endpoint pair addressing. There is a correspondence between endpoints 1 to 3 and endpoints 4 to 6, e.g. endpoint 1 pair addressing is enabled, endpoint 4 becomes the pair endpoint. In this case, the endpoint 1 should be configured as an OUT endpoint, while endpoint 4 is an IN endpoint.

This correspondence is transparent to the USB host which considers that there are two endpoints at address 0x01, an OUT.

USB_RDMA_LEN: Receive DMA Packet Length Requested

addr 500001Ch R/W 16 bits

- **Bits 15..9 – Reserved**
- **Bits 8..0 – URDL[8:0]: USB Receive DMA Length**

Note: Default: 00h

ARM programs this register with the number of bytes to be transferred during the next DMA.

USB_DMA_FADD: DMA Target Endpoint Address

addr 5000020h R/W 16 bits

- **Bits 7..0 – FAD[7:0]: USB Target Endpoint FIFO Address**

Note: Default: 00h

This register is programmed with the address of the Endpoint FIFO Register of the USB block that the DMA operation is going to transfer bytes from/to. The addresses of the six endpoints supported by the USB block are listed in Table 5.

Table 5. DMA Target Endpoint Addresses

Address	Target
0xCF	FDR0 Function Endpoint0
0xCE	FDR1 Function Endpoint1
0xCD	FDR2 Function Endpoint2
0xCC	FDR3 Function Endpoint3
0xCB	FDR4 Function Endpoint4
0xCA	FDR5 Function Endpoint5
0xC9	FDR6 Function Endpoint6

USB_RDMA_LENR: Receive DMA Packet Length Transferred

addr 5000024h R 8 bits

- **Bits 7..0 URDL[7:0]: Receive DMA Transferred Length**

Note: Default: 00h

The contents of this register after the end of a DMA reflect the number of bytes that have been transferred.

USB_RDMA_EN

addr 5000028h R/W 8 bits

Bits 7..2 – Reserved

- **Bit 1 – RDMAEN**

Activates receive DMA (DMA for an OUT endpoint). This bit is reset after the completion of the DMA.

- **Bit 0 – Reserved**

Note: Default: 00h

USB_DMA_RADD: RAM Target Address for DMA Cycles

addr 5000040h R/W 32 bits

- **Bits 31..24 – Reserved**
- **Bits 23..0 – RAD[23:0]**

Target address in SRAM for DMA transfers.

Note: Default: 00h

USB_TDMA_LEN: Transmit DMA Packet Length Requested

addr 5000044h R/W 16 bits

- **Bits 15..9 – Reserved**
- **Bits 8..0 – UTDL[8:0]: USB Transmit DMA Length**

Note: Default: 00h

The ARM programs this register with the number of bytes to be transferred during the next DMA.

USB_TDMA_LENR: Transmit DMA Packet Length Transferred

addr 5000048h R 8 bits

- **Bits 7..0 UTDL[7:0]: Transmit DMA Transferred Length**

Note: Default: 00h

After the end of a DMA, the contents of this register reflect the number of bytes that have been transferred from main system memory to the transmit FIFO.

USB_TDMA_EN: Transmit DMA Enable

addr 500004Ch R/W 8 bits

Bits 7..2 – Reserved

- **Bit 1 – Reserved**
- **Bit 0 – TDMAEN**

Activates transmit DMA (DMA for an IN endpoint)

Note: Default: 00h

This bit is reset after the completion of the DMA.

FRM_NUM_H: Frame Number High Register

addr 50003F4h W 8 bits

Bits 7..3 – Reserved

- **Bits 2..0 – FCH[10:8]**

This is the upper 3 bits of the 11-bit frame number of SOF packet.

Note: Default: 00h

FRM_NUM_L: Frame Number Low Register

addr 50003F0h W 8 bits

Bits 7..3 – Reserved

- **Bits 2..0 FCL[7:0]**

This is the lower 8 bits of the 11-bit frame number of SOF packet.

Note: Default: 00h

GLB_STATE: Global State Register

addr 50003ECh 8 bits

- **Bits 7..4 – Reserved**
- **Bit 3 W – RSMINPR**

Set by the hardware when a Resume is send in the USB bus during Remote Wake-up feature (13 ms).

- **Bit 2 R – RMWUPE**

Remote Wake-up Enable. This bit is set if the Host enables the function's remote wake-up feature.

- **Bit 1 R – CONFG**

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.

- **Bit 0 R – FADD Enable: Function Address Enable**

This bit is set by firmware after the status phase of a SET_ADRESS request transaction. The Host will use the new address starting at the next transaction.

Note: Default: 00h

SPRSR: Suspend/Resume Register

addr 50003E8h W 8 bits

- **Bits 7..4 – Reserved**
- **Bit 3 – SOF INT: Start Of Frame Interrupt**
- **Bit 2 – EXT RSM: Received External Resume**

The USB hardware sets this bit to denote an External Resume Interrupt. If RMWUPE = 1, a RESUME signal is send in USB BUS.

- **Bit 1 – RCVD RSM: Received Resume**

The USB hardware sets this bit when a USB resume signaling is detected at its port.

- **Bit 0 – SUSP: 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.

Note: Default: x0h

SPRSIE: Suspend/Resume Interrupt Enable Register

addr 50003E4h R 8 bits

Bits 7..4 – Reserved

- **Bit 3 – SOF IE: Enable SOF Interrupt**
- **Bit 2 – EXTRSM IE: Enable External Resume Signaling Interrupt**

1 = Enable

0 = Disable

- **Bit 1 – RCVDRSM IE: Enable BUS Resume Signaling Interrupt**

1 = Enable

0 = Disable

- **Bit 0 – SUSP IE: Enable Suspend Signaling Interrupt**

1 = Enable

0 = Disable

Note: Default: x0h

UISR: USB Interrupt Status Register

addr 50003DChW8 bits

- **Bit 7 – Reserved**
- **Bit 6 – EP6 INT: Endpoint 6 Interrupt**
- **Bit 5 – EP5 INT: Endpoint 5 Interrupt**
- **Bit 4 – EP4 INT: Endpoint 4 Interrupt**
- **Bit 3 – EP3 INT: Endpoint 3 Interrupt**
- **Bit 2 – EP2 INT: Endpoint 2 Interrupt**
- **Bit 1 – EP1 INT: Endpoint 1 Interrupt**
- **Bit 0 – EP0 INT: Endpoint 0 Interrupt**

Note: Default: 00h

The function interrupt bits will be set by the hardware whenever the following bits in the corresponding Endpoint's Control and Status Register are modified by the USB hardware:

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

UIAR: USB Interrupt Acknowledge Register

addr 50003D4h W 8 bits

- **Bit 7 – Reserved**
- **Bit 6 – EP6 INTA: Endpoint 6 Interrupt Acknowledge**
- **Bit 5 – EP5 INTA: Endpoint 5 Interrupt Acknowledge**
- **Bit 4 – EP4 INTA: Endpoint 4 Interrupt Acknowledge**
- **Bit 3 – EP3 INTA: Endpoint 3 Interrupt Acknowledge**
- **Bit 2 – EP2 INTA: Endpoint 2 Interrupt Acknowledge**
- **Bit 1 – EP1 INTA Endpoint 1 Interrupt Acknowledge**
- **Bit 0 – EP0 INTA: Endpoint 0 Interrupt Acknowledge**

Note: Default: 00h

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.

UIER: USB Interrupt Enable Register

addr 50003CCh R/W 8 bits

- **Bit 7 – SOFIE: Enable SOF Interrupt**
- **Bit 6 – EP6 IE: Enable Endpoint 6 Interrupt**
- **Bit 5 – EP5 IE: Enable Endpoint 5 Interrupt**
- **Bit 4 – EP4 IE: Enable Endpoint 4 Interrupt**
- **Bit 3 – EP3 IE: Enable Endpoint 3 Interrupt**
- **Bit 2 – EP2 IE: Enable Endpoint 2 Interrupt**
- **Bit 1 – EP1 IE: Enable Endpoint 1 Interrupt**
- **Bit 0 – EP0 IE:Enable Endpoint 0 Interrupt**

Note: Default: 00h

The bits in this register has the following meaning:

1 = Enable interrupt

0 = Disable interrupt

FADDR: Function Address Register

addr 50003C8h R 8 bits

- **Bit 7 – FEN: Function Enable**
- **Bits 6..0 – FADD[6:0]: Function Address**

Note: Default: 00h

The FIU address register contains the function address assigned by the Host. This Function Address Register must be programmed by the processor once it has:

1. Received a SET_ADDRESS command from the Host.
2. 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.

ENDPPGPG: Endpoint Ping-pong Enable Register

addr 50003C4h R 8 bits

Bit 7 – Reserved

- **Bit 6 – PG_EP6_EN:Enable Endpoint 6 Ping-pong**
- **Bit 5 – PG_EP5_EN: Enable Endpoint 5 Ping-pong**
- **Bit 4 – PG_EP4_EN: Enable Endpoint 4 Ping-pong**
- **Bit 3 – PG_EP3_EN: Enable Endpoint 3 Ping-pong**
- **Bit 2 – PG_EP2_EN: Enable Endpoint 2 Ping-pong**
- **Bit 1 – PG_EP1_EN: Enable Endpoint 1 Ping-pong**
- **Bit 0 – PG_EP0_EN: Enable Endpoint 0 Ping-pong**

Endpoint Control Registers

addr: see below 8 bits

Endpoint Control Registers

addr: see below 8 bits

- **Bit 7 R – EPEDS: Endpoint Enable/Disable**

0 = Disable Endpoint

1 = Enable Endpoint

- **Bit 6 – Reserved**
- **Bits 5..4 – Reserved and set to 0**
- **Bit 3 W – DTGLE: Data Toggle**

Identifies DATA0 or DATA1 packets.

- **Bit 2 R – EPDIR: Endpoint Direction**

Only applicable for non-control Endpoints (0 = Out, 1 = In).

- **Bits 1..0 R – EPTYPE: Endpoint Type**

These bits represent the type of the Endpoint (see Tables 6, 7, 8 and 9).

Table 6. Endpoint Type

Bit 1	Bit 0	Type
0	0	Control
0	1	Isochronous
1	0	Bulk
1	1	Interrupt

Note: Default: 00h

Table 7. Endpoint Control Registers' Address

Address	Register
D0003BCh	ENDP0_CNTR Endpoint 0
D0003B8h	ENDP1_CNTR Endpoint 1
D0003B4h	ENDP2_CNTR Endpoint 2
D0003B0h	ENDP3_CNTR Endpoint 3
D0003ACh	ENDP4_CNTR Endpoint 4
D0003A8h	ENDP5_CNTR Endpoint 5
D0003A4h	ENDP6_CNTR Endpoint 6

Table 8. Endpoint Control and Status Register

addr: see below 8 bits

Bit	Bit Name	Function	Description of Bit
Bit 7 R	Control Direction	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.	This bit is only used by Control Endpoints. It is used by firmware to indicate the direction of a control transfer. It is written by the firmware after it receives a RX SETUP interrupt. The hardware uses this bit to determine the status phase of a control transfer.
Bit 6 R	Data End	Indicates that the processor has placed the last data packet in FIFO0, or that the processor has processed the last data packet it expects from the Host.	This bit is used only by Control Endpoints. Together with bit 1 (TX Packet Ready) it will signal the USB hardware to go to the STATUS phase after the packet currently residing in the FIFO is transmitted. After the 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. For this reason, Tx_Packet Ready should be set before Data_End.
Bit 5 R	Force Stall	Set by the processor to indicate a stalled Endpoint. The hardware will send a STALL handshake as a response to the next IN or OUT token.	The processor sets this bit if it wants to force a STALL after an unsupported request is received, or if the Host continues to ask for data after the data is exhausted. This bit should be set at the end of any data phase or setup phase.
Bit 5 R/C	TX Packet Ready	Indicates that the processor has loaded the FIFO with a packet of data. This bit is cleared by hardware after the USB Host acknowledges the packet. For ISO Endpoints, this bit is cleared unconditionally after the data is sent.	This bit is used for the following operations: Control read transactions by a Control Endpoint. IN transactions with DATA1 PID to complete the status phase for a Control Endpoint, when this bit is "0”, but bit Data End (bit 4) is "1”. By a BULK IN or ISO IN or INT IN Endpoint. The processor should write to the FIFO only if this bit is cleared. The bit will be set after it has completed writing the data. The data can be a zero length. For a Control Endpoint the processor should write to the FIFO only while bit 6 (TX Packet Requested) is set. The hardware clears this bit after it receives an ACK. If the interrupt is enabled, clearing this bit by the hardware, it causes an interrupt to the processor.
Bit 3 W	Stall Snd	The USB hardware sets this bit after a STALL is sent to the Host. Indicates End of data stage for the Control Endpoint only.	The USB hardware sets this bit after a STALL has been sent. The firmware uses this bit when responding to a USB GetStatus Request.

Table 8. Endpoint Control and Status Register (Continued)

addr: see below 8 bits

Bit	Bit Name	Function	Description of Bit
Bit 2 W	RX Setup	The USB hardware sets this bit when it receives a valid setup packet from the Host. This bit is used by Control Endpoints only.	This bit is used only by Control Endpoints to signal 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 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 data has been completely read from the FIFO the firmware should clear this bit.
Bit 1 W	RX OUT Packet	Indicates that the USB hardware has decoded an OUT token and that the data is in the FIFO.	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 hardware will NAK all OUT tokens. For Control Endpoints only, bit 7 of this register, Enable Control Write, has to be set for the 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: Control write transactions by a Control Endpoint OUT transaction with DATA1 PID to complete the status phase of a control Endpoint. 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.
Bit 0 R	TX Complete	The hardware sets this bit to indicate to a Control Endpoint that it has received an ACK handshake from the Host.	This bit is used by 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: Control read data stage Status stage without data stage Status stage after a control write transaction

Note: Default: 00h

Table 9. Endpoint Control and Status Register's Address

Address	Register
D00037Ch	FCSR0 Endpoint 0
D000378h	FCSR1 Endpoint 1
D000374h	FCSR2 Endpoint 2
D000370h	FCSR3 Endpoint 3
D00036Ch	FCSR4 Endpoint 4
D000368h	FCSR5 Endpoint 5
D000364h	FCSR6 Endpoint 6

UART Register Set

In Table 10, the register file and its fields are briefly presented. A more detailed description is provided in the following sections.

Table 10. UART Register File and Register Fields

Addr. (hex)	Register	Bit-7	Bit-6	Bit-5	Bit-4	Bit-3	Bit-2	Bit-1	Bit-0
700000	US_RHR	X	X	X	X	X	X	X	X
700000	US_THR	X	X	X	X	X	X	X	X
700004	US_IER	Transmitter empty interrupt	Receive time-out interrupt	Parity error interrupt	Framing error interrupt	Overrun error interrupt	Receive break interrupt	Transmit holding register	Receive holding register
700008	US_FCR	RCVR trigger (MSB)	RCVR trigger (LSB)	0	0	DMA mode select	0	RCVR FIFO reset	FIFO enable
70000C	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
700010	US_MR	Channel mode (CHM1)	Channel mode (CHM0)	0	0	0	0	0	0
700014	US_CSR	Transmitter empty	Receive time-out	Parity error	Framing error	Overrun error	Receive break	Transmit holding register ready	Receive holding register ready
700018	US_CR	Rx enable	Reset status bit	Tx enable	Restart time-out	Tx teset	Rx teset	Stop break	Start break
70001C	US_BL	MSB							LSB
700020	US_BM	MSB							LSB
700024	US_RTO	MSB							LSB
700028	US_TTG	MSB							LSB
70002C	US_MCI	DSR	RI	CD	CTS	DSR change	RI change	CD change	CTS change
70002C	US_MCO	0	0	0	0	0	0	RTS	DTR
700030	US_MCC	0	LB RI value	LB CD value	MC Loop back enable	DSR change mask	RI change mask	CD change mask	CTS change mask

US_RHR: Receive Holding Register

addr: 700000 hex R 8 bits

- **Bits 7..0 – US_RHR[7:0]**

Received data

Note: Default Value: 00 hex

US_THR: Transmit Holding Register

addr: 700000 hex W 8 bits

- **Bits 7..0 – US_THR[7:0]**

Transmit data

Note: Default Value: 00 hex

US_IER: Interrupt Enable Register

addr: 700004 hex R/W 8 bits

- **Bit 7 – TXEI: 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 Status Register SREG of the MCU is set, an interrupt will occur.

- **Bit 6 – RXTOI: 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 – PEI: Parity Error Interrupt**

This bit when set enables the Parity Error interrupt. If the Parity Error bit of Control Status Register US_CSR is set, an interrupt will occur.

- **Bit 4 – FEI: Framing Error Interrupt**

This bit when set enables the Framing Error interrupt. If the Parity Error bit of Control Status Register US_CSR is set, an interrupt will occur.

- **Bit 3 – OEI: Overrun Error Interrupt**

This bit when set enables the Overrun Error interrupt. If the Parity Error bit of Control Status Register US_CSR is set, an interrupt will occur.

- **Bit 2 – RBRI: 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 – THRI: Transmit 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 – RHRI: 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 SREG of the MCU are set.

Note: Default Value: 00 hex – The Modem Control Interrupts are controlled only from the US_MCC register.

US_FCR: FIFO Control Register

addr: 700008 hex R/W 8 bits

- **Bits 7..6 RCVR[1:0]a; RCVR Trigger Bits**

These bits indicate the minimum number of bytes required in the receive FIFO to generate a receive ready interrupt.

- **Bits 5..4 – Reserved**
- **Bit 3 – RDMA: 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 – FRS: FIFO Reset**

When set, resets the receive FIFO.

- **Bit 0 – FEN: FIFO Enable**

When set, enables the 16 byte receive FIFO.

Note: Default Value: 00 hex

The trigger level is shown in Table 11.

Table 11. Trigger Level

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

Protocol Mode Register (US_PMR)

US_PMR: Protocol Mode Register

addr: 70000C hex R/W 8 bits

- **Bits 7..6 – CHL[1:0]: Character Length**

00: 5 bits

01: 6 bits

10: 7 bits

11: 8 bits

- **Bits 5..4 – SBN[1:0]: Number of Stop Bits**

00: 1-bit time

01: 1- ,5-bit time

10: 2-bit time

11: Reserved

- **Bits 3..2 – PM[1:0]: Parity Mode**

00: Normal parity

01: Parity forced

10: No parity

11: Multi drop

- **Bit 1 – RES: 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 – LSB: Reserved**

Note: Default Value: 00 hex

US_MR: Mode Register

addr: 700010 hex R/W 8 bits

- **Bits 7..6 – CHM[1:0] Channel Mode**

00: Normal

01: Automatic echo

10: Local loop-back

11: Remote loop-back

- **Bits 5..0 – Reserved**

Note: Default Value: 00 hex

US_CSR: Control Status Register

addr: 700014 hex R/W 8 bits

- **Bit 7 – TXE: Transmitter Empty**

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

- **Bit 6 – RXTO: Receive Time-out**

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

- **Bit 5 – PE: Parity Error**

When set, this indicates that a parity error has occurred.

- **Bit 4 – FE: Framing Error**

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

- **Bit 3 – OE: 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 – RBR: Receive Break**

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

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

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

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

When set indicates that the Receive Holding Register is full. In order to clear this bit you must empty the RHR (or the FIFO if it is enabled) by reading the US_RHR register.

Note: Default Value: 00 hex

US_CR: Control Register

addr: 700018 hex R/W 8 bits

- **Bit 7 – RXEN: Enable**

When set, this enables the receiver block of UART.

- **Bit 6 – RLES: Reset Line Error Status bits**

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

- **Bit 5 – TXEN: Tx Enable**

When set, this enables the transmitter block of UART.

- **Bit 4 – RSTO: Restart Time-out**

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

- **Bit 3 – TXRS: Tx Reset**
- **When set, this resets the transmit logic.**
- **Bit 2 – RXRS: Rx Reset**

When set, this resets the receive logic.

- **Bit 1 – SPB: Stop Break**

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

- **Bit 0 – STB: Start Break**

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

Note: Default Value: 00 hex

US_BL: Low Byte, Baud Rate Register

addr: 70001C hex R/W 8 bits

- **Bits 7..0 – US_BL[7:0]**

Baud rate generator division ratio low. The main system clock is divided by the number contained in US_BL and US_BM, to provide the USART clock (which is 16 times the actual serial data rate).

Note: Default Value: 00 hex

US_BM: High Byte, Baud Rate Register

addr: 700020 hex R/W 8 bits

- **Bits 7..0 – US_BM[7:0]**

Baud rate generator division ratio high.

Note: Default Value: 00 hex

Table 12. Baud Rate Generation Example (Internal UART Clock = 14,76923 MHz)

Baud Rate	US_BM (hex)	US_BL (hex)	Error %
100	24	00	0.16
200	12	00	0.16
400	09	00	0.16
600	06	00	0.16
1200	03	00	0.16
2400	01	80	0.16
4800	00	C0	0.16
9600	00	60	0.16
19200	00	30	0.16
28800	00	20	0.16
38400	00	18	0.16
57.6K	00	10	0.16
115.2K	00	08	0.16
230.4K	00	04	0.16
307.2K	00	03	0.16
460.8K	00	02	0.16
921.6K	00	01	0.16

US_RTO: Receiver Time-out Register

addr: 700024 hex R/W 8 bits

- **Bits 7..0 – US_RTO[7:0]**

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

Note: Default Value: 00 hex

US_TTG: Transmitter Time Guard Register

addr: 700028 hex R/W 8 bits

- **Bits 7..0 – US_TTG[7:0]**

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

Note: Default Value: 00 hex

US_MCI Modem Control Inputs

addr: 70002C hex R 8 bits

- **Bit 7 – DSR: Data Set Ready**

Active high. This bit is the compliment of the DSR input pin.

- **Bit 6 – RI: Ring Indicator**

Active high. This bit is the compliment of the RI input pin.

- **Bit 5 – CD: Carrier Detect**

Active high. This bit is the compliment of the CD input pin.

- **Bit 4 – CTS: Clear To Send**

Active high. This bit is the compliment of the CTS input pin.

- **Bit 3 – DSR Change**

Active high

Logic 0: No DSR change

Logic 1: The DSR input pin has changed state since the last time it was read. An interrupt will be generated.

- **Bit 2 – RI: Ring Indicator Change**

Active high

Logic 0: No RI change

Logic 1: The RI input pin has changed state since the last time it was read. An interrupt will be generated.

- **Bit 1 – CD: Change**

Active high

Logic 0: No CD change

Logic 1: The CD input pin has changed state since the last time it was read. An interrupt will be generated.

- **Bit 0 – CTS Change**

Active high

Logic 0: No CTS change

Logic 1: The CTS input pin has changed state since the last time it was read. An interrupt will be generated.

Note: Default Value: 00 hex

US_MCO: Modem Control Outputs

addr: 70002C hex W 8 bits

- **Bits 7..5 – Reserved**
- **Bit 1 – RTS**

Control RTS output pin:

Logic 1: Force RTS output pin to a logic

Logic 0: Force RTS output pin to a logic 0

- **Bit 0 – DTR**

Control DTR output pin:

Logic 1: Force DTR output pin to a logic 1

Logic 0: Force DTR output pin to a logic 0

Note: Default Value: 00 hex

US_MCC: Modem Control Register

addr: 700030 hex R/W 8 bits

- **Bit 7 – Reserved**
- **Bit 6 – LB RI Value**

The compliment of the value of RI input, when Modem Control Loop Back mode is enabled.

- **Bit 5 – LB CD Value**

The compliment of the value of CD input, when Modem Control Loop Back mode is enabled.

- **Bit 4 – MC LB EN: Modem Control Loop Back Mode Enable**
- **Bit 3 – DSR Change Mask**

This bit when set enables the DSR change interrupt. If the DSR change bit of Modem Control Input Register US_MCI is set, then an interrupt will occur.

- **Bit 2 – RI Change Mask**

This bit when set enables the RI change interrupt. If the RI change bit of Modem Control Input Register US_MCI is set, then an interrupt will occur.

- **Bit 1 – CD Change Mask**

This bit when set enables the CD change interrupt. If the CD change bit of Modem Control Input Register US_MCI is set, then an interrupt will occur.

- **Bit 0 – CTS Change Mask**

This bit when set enables the CTS change interrupt. If the CTS change bit of Modem Control Input Register US_MCI is set, then an interrupt will occur.

Note: Default Value: 00 hex

The low nibble of the Modem-control register is the Modem Control interrupt mask register (see US_MCI).

Note: US_IER cannot control the interrupts from modem control logic.

The high nibble is used to control the Modem Control Loop-back mode. If MC Loop Back enable bit is set, then the US_MCC bits 5 - 6 control the values of CD and RI inputs (see and US_MCI bits 5 - 6 and 1 - 2).

When MC Loop-back mode is enabled, then input pins RI, CD, DSR, CTS are assigned the following values:

RI bit 6 of register US_MCC (inverted)

CD bit 5 of register US_MCC (inverted)

DSR pin DTR

CTS pin RTS

General Purpose Registers

Power Down

addr: D00000 hex R/W 6 bits

- **Bit 5 – VCPU: VC Power Up**
- **Bit 4 – UARTPU: UART Power Up**
- **Bit 3 – USBPU: USB Power Up**
- **Bit 2 – BDPRPD: BT Baseband Power Down**
- **Bit 1 – Reserved**
- **Bit 0 – ARMSPD: ARM Speed**

Note: Default Value: 00 hex

Module Reset

addr: D00004 hex R/W 5 bits

- **Bit 4 – PCMCIARES: PCMCIA Core Reset**
- **Bit 3 – USBRES: USB Reset**
- **Bit 2 – VCRES: VC Reset**
- **Bit 1 – UARTRES: UART Reset**
- **Bit 0 – BTRES: BT Baseband Reset**

Note: Default Value: 00 hex

USB Pad Enable

addr: D00008 hex R/W 1 bit

- **Bit 2 – USBPDEN: USB Pad Enable**

Note: Default Value: 0

DEBUG Port

addr: D0000C hex R/W 1 bit

- **Bit 9 – BTRXDB**

When enabled, the output of the BT clock recovery circuit is transferred to the Debug pins.

- **Bit 8 DBEN: Debug TX Enable**
- **Bit 7..0 – DBDATA: Debug Data**

Interrupt Controller Registers

Interrupt Status Register

addr: F00000 hex R 8 bits

- **Bit 7 – VC: Voice Codec Interrupt**
- **Bit 6 – PCMCIA:PCMCIA Interrupt**
- **Bit 5 – UART: UART Interrupt**
- **Bit 4 – USB: USB Interrupt**
- **Bit 3 – BDPR: Baseband DPRAM Interrupt**
- **Bit 2 – BT: Baseband and Timers Interrupt**
- **Bit 1 – TM1: Timer 1 Interrupt**
- **Bit 0 – TM2: Timer 2 Interrupt**

Note: Default Value: 00 hex

Interrupt Mask Register

addr: F00004 hex R/W 8 bits

- **Bit 7 – VC Mask**

Enables the Voice Codec interrupt

- **Bit 6 – PCMCIA Mask**

Enables the PCMCIA interrupt

- **Bit 5 – UART Mask**

Enables the UART interrupt

- **Bit 4 – USB Mask**

Enables the USB interrupt

- **Bit 3 – BDPR Mask**

Enables the Baseband DPRAM interrupt

- **Bit 2 – BT Mask**

Enables the Baseband and Timers interrupt

- **Bit 1 – TM1 Mask**

Enables the Timer 1 Interrupt

- **Bit 0 – TM2 Mask**

Enables the Timer 2 Interrupt

Note: Default Value: 00 hex

Interrupt Masked Status Register

addr: F00008 hex R 8 bits

- **Bits 7..0 – US_ISM[7:0]**

Each bit of this register is the boolean and of the corresponding bits of Interrupt Status register and the Interrupt Mask Register.

Note: Default Value: 00 hex

Interrupt Priority Register

addr: F0000C hex R/W 8 bits

- **Bit 7 – VC_PR**

Enables FIQ priority for the Voice Codec interrupt line

- **Bit 6 – PCMCIA_PR**

Enables FIQ priority for PCMCIA interrupt line

- **Bit 5 – UART_PR**

Enables FIQ priority for UART interrupt line

- **Bit 4 – USB _PR**

Enables FIQ priority for USB interrupt line

- **Bit 3 – BDPR_PR**

Enables FIQ priority for DPRAM interrupt line

- **Bit 2 – BT_PR**

Enables FIQ priority for Baseband interrupt line

- **Bit 1 – TM1_PR**

Enables FIQ priority for the Timer 1 interrupt line

- **Bit 0 – TM2_PR**

Enables FIQ priority for the Timer 2 interrupt line

Note: Default Value: 00 hex

Timer Device Registers

AT76C551 incorporates two identical and completely independent system timer devices, Timer Device 1 and Timer Device 2.

Each Timer Device is implemented as a 32-bit down counter which advances at a programmable rate driven by a prescale circuit. The counter is loaded with a programmable 32-bit value (preload) when the Timer Device is enabled. When the counter reaches zero an interrupt to the ARM core is generated. Timer Device behavior after the counter reaches zero depends on the selected operating mode. In one-shot mode the counter stops and no interrupt is generated to the ARM until the Timer Device is reenabled. In periodic mode the counter is reloaded with the preload value and countdown is continued, effectively generating a periodic interrupt to the ARM core.

AT76C551 firmware can program and control each Timer Device via registers mapped into AMBA memory space and thus accessible by the ARM core. (See Table 13).

Table 13. Timer Device Registers

Register Name	AMBA Address for Timer Device 1	AMBA Address for Timer Device 2
Timer Preload Low Register	F200000h	F200020h
Timer Preload High Register	F200004h	F200024h
Timer Value Low Register	F200008h	F200028h
Timer Value High Register	F20000Ch	F20002Ch
Timer Prescale Register	F200010h	F200030h
Timer Control Register	F200014h	F200034h
Timer Interrupt Clear Register	F200018h	F200038h

Timer Preload Low Register

R/W 16 bits

Timer Preload High Register

R/W 16 bits

Timer Preload Low Register and Timer Preload High Register contain the 32-bit preload value.

Timer Value Low Register

R 16 bits

Timer Value High Register

R 16 bits

Timer Value Low Register and Timer Value High Register reflect the 32-bit counter current value. Note that full 32-bit counter value can be acquired by reading both Timer Value Low Register and Timer Value High Register, but result consistency cannot be guaranteed by the Timer Device and must be ensured by software means.

Timer Prescale Register

R/W 16 bits

Timer Prescale Register sets the divisor for the prescale circuit. If Timer Prescale Register contains p and PAI clock frequency is f (MHz) then countdown rate for the Timer Device counter will be r = f/p

Note: The Timer Prescale Register must contain a non-zero value for proper Timer Device operation.

Timer Control Register

R/W 16 bits

- **Bits 15..2 – Reserved**

Returns 0 when read

- **Bit 1 – Enable**

Logic 1: The counter is allowed to run.

Logic 0: The counter is stopped. The Timer Device may be configured via Timer Preload Low Register, Timer Preload High Register and Timer Prescale Register.

- **Bit 0 – Periodic**

Logic 1: Periodic mode operation is selected.

Logic 0: One-shot mode operation is selected.

Timer Interrupt Clear Register

W 16 bits

Writing Timer Interrupt Clear Register with any value clears an interrupt generated by the Timer Device.

AT76C551 firmware is expected to write to Timer Interrupt Clear Register each time a Timer Device interrupt is processed or else, interrupt request to the ARM core will persist.

Electrical Specifications

Recommended Operating Conditions

Table 14 shows the range for which Atmel library cells have been characterized. Operation of a device outside this range may result in the device failing to meet some of its specification.

Table 14. Recommended Operating Conditions

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
V_{DD3}	DC Supply Voltage	Core and Standard I/Os	3.0	3.3	3.6	V
V_I	DC Input Voltage		0		VDD	V
V_O	DC Output Voltage		0		VDD	V
TEMP	Operating Free Air Temperature Range	Industrial	-40		+85	°C

Absolute Maximum Ratings

Operation of a device outside of the range givein in Table 15 may cause permanent damage to the device and/or affect reliability.

Table 15. Absolute Maximum Ratings

Symbol	Parameter	Conditions	Min	Max	Unit
V_{DD3}	DC Supply Voltage	Core and Standard I/Os	-0.3	3.6	V
V_I	DC Input Voltage		-0.3	VDD+0.3	V
V_O	DC Output Voltage		-0.3	VDD+0.3	V
TEMP	Operating Free Air Temperature Range	Industrial	-40	+85	°C
TSG	Storage Temperature		-60	+150	°C

DC Characteristics

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

Oscillator Signals: XIN, XOUT

Table 16. Power Supply

Symbol	Parameter	Condition	Min	Max	Unit
V_{CC}	Power Supply			3.3	V
I_{CC}	Supply Current			50	mA
I_{CCS}	Suspended Device Current			200	µA

Table 17. USB Signals: DP, DM

Symbol	Parameter	Condition	Min	Max	Unit
I_{LO}	High-Z Data Line Leakage	0V < Vin < 3.3V	-10	+10	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 kΩ to 3.6V		0.3	V
V_{OH1}	Static Output High	RL of 15 kΩ to GND			

Table 18. Oscillator Signals: XTAL1, XTAL2[1]

Symbol	Parameter	Condition	Min	Max	Unit
V_{LH}	XTAL1 Switching Level		0.47	1.20	V
V_{HL}	XTAL2 Switching Level		0.67	1.44	V
C_{X1}	Input Capacitance, XTAL1			9	pF
C_{X2}	Output Capacitance, XTAL2			9	pF
C_{12}	XTAL1/XTAL2 Capacitance			1	pF
t_{SU}	Start-up Time	6 MHz, Fundamental		2	ms
D_L	Drive Level	V_{CC} = 3.3V, 13 MHz crystal, 120Ω Equiv. Series Resistor		1	mW

Note: 1. XTAL2 must not be used to drive other circuitry.

AC Characteristics

Table 19. DP, DM Driver Characteristics

Symbol	Parameter	Condition	Min	Max	Unit
t_R	Rise Time	C_L = 50 pF	4	20	ns
t_F	Fall Time	C_L = 50 pF	4	20	ns
t_{RFM}	TR/TF Matching		90	110	%
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	44	W

Note: 1. With external 27W series resistor

Table 20. DP, DM Data Source Timings

Symbol	Parameter	Condition	Min	Max	Unit
T_{DRATE}	Full Speed Data Rate	Average Bit Rate	11.97	12.03	Mbs
T_{FRAME}	Frame Interval		0.9995	1.0005	ms
T_{DJ1} T_{DJ2}	Source Diff Driver Jitter To Next Transition For Paired Transitions		-3.5 -4.0	3.5 4.0	ns ns
T_{FDEOP}	Source Jitter for Differential Transition to SE0 Transition		-2	5	ns
T_{FEOPT}	Source SE0 interval of EOP		160	175	ns
T_{FEOPR}	Receiver SE0 interval of EOP		82		ns
T_{JR1} T_{JR2}	Recvr Data Jitter Tolerance To Next Transition For Paired Transitions		-18.5 -9	18.5 9	ns ns
T_{FST}	Width of SE0 interval during differential transition			14	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

© Atmel Corporation 2001.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

ATMEL® is the registered trademarks of Atmel.

Bluetooth™ is a trademark owned by the Bluetooth SIG, Inc.; ARM7TDMI®, ARM®, Thumb® are the registered trademarks of ARM Limited; AMBA™ is the trademark of ARM Limited.

Other terms and product names may be the trademarks of others.

Printed on recycled paper.

1612D–08/01/xM