Chipsmall

Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!

Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832
Email & Skype: info@chipsmall.com Web: www.chipsmall.com
Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China

Features

- **AVR® 8-bit RISC Microcontroller with 83 ns Instruction Cycle Time**
- **USB Hub with One Attached and Four External Ports**
- **USB Function with Two Programmable Endpoints**
- **External Program Memory, 512-byte Data SRAM**
- **32 x 8 General Purpose Working Registers**
- **32 Programmable I/O Port Pins**
- **Programmable Serial UART**
- **Master/Slave SPI Serial Interface**
- **One 8-bit Timer/Counter with Separate Pre-scaler**
- **One 16-bit Timer/Counter with Separate Pre-scaler and Two PWMs**
- **External and Internal Interrupt Sources**
- **Programmable Watchdog Timer**
- **6 MHz Oscillator with On-chip PLL**
- **5V Operation with On-chip 3.3V Power Supply**
- **100-lead LQFP Package**

Description

The Atmel AT43USB320A is an 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the AT43USB320A achieves throughputs approaching 12 MIPS. The AVR core combines a rich instruction set with 32 general-purpose working registers. All 32 registers are directly connected to the ALU allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The AT43USB320A features an on-chip 512-byte of data memory. It is supported by a standard set of peripherals such as timer/counter modules, watchdog timer and internal and external interrupt sources. The major peripheral included in the AT43USB320A is the USB Hub with an embedded function for use in peripherals such as monitor with remote control as shown in Figure 1.

Note: There are two versions of the AT43USB320A. They are indicated by the internal part numbers 55618D and 55618E. The only difference between the two versions is in the polarity of the SUSPEND pin. The 55618D SUSPEND pin is active low, while the 55618E SUSPEND pin is active high.

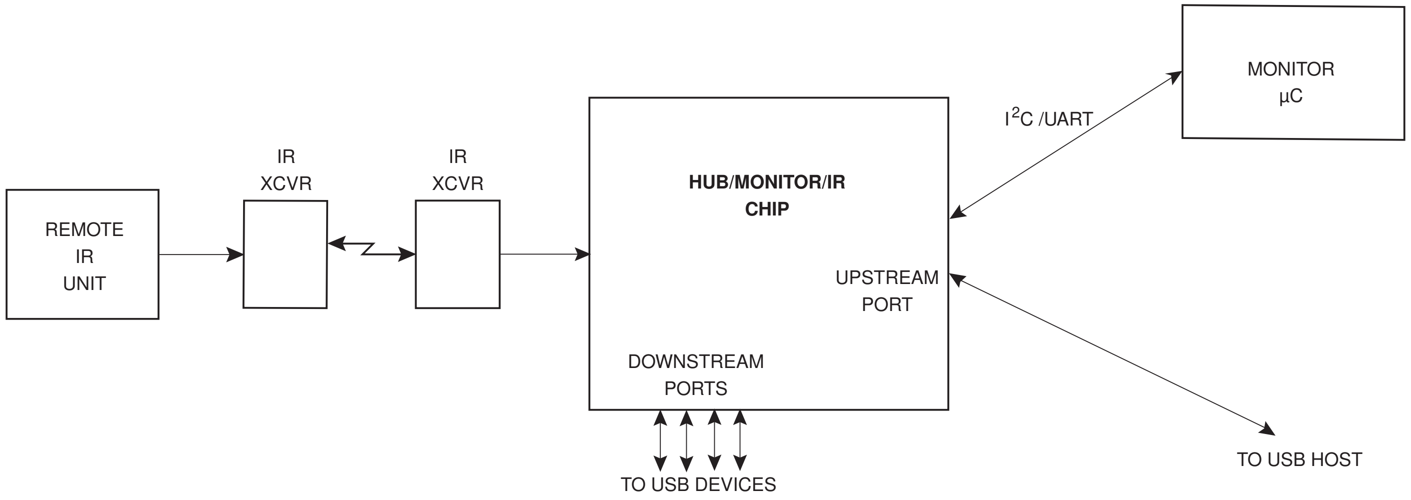
Full-speed USB Microcontroller with an Embedded Hub

AT43USB320A

Rev. 1443D–USB–12/03

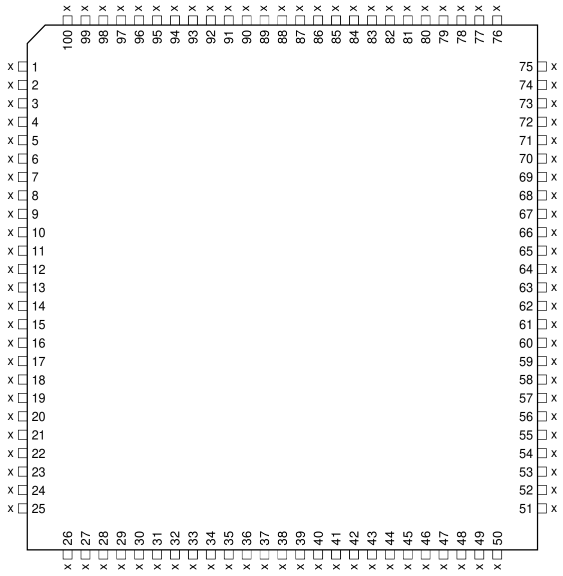
Hub/Monitor/IR Chip Application

Figure 1. Application Example



Pin Configurations

100-lead LQFP



Pin Assignment

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

Pin Number	Signal	Type
1	PD2	B
2	PD3	B
3	PD4	B
4	PD5	B
5	PD6	B
6	PD7	B
7	6/12N	I
8	LFT	O
9	XTAL1	I
10	XTAL2	O
11	VSS	V
12	TESTN	I
13	A0	B
14	A1	B
15	A2	B
16	A3	B
17	A4	B
18	A5	B
19	A6	B
20	A7	B
21	VSS	V
22	A8	B
23	A9	B
24	A10	B
25	NC	–
26	NC	–
27	A11	B
28	A12	B
29	A13	B
30	A14	B
31	A15	B
32	VCC	V
33	VSS	V
34	CEXT1	O
35	SUSPEND	O
36	D0	I
37	D1	I
38	D2	I
39	D3	I
40	D4	I
41	D5	I
42	D6	I
43	D7	I
44	VSS	V
45	D8	I
46	D9	I
47	D10	I
48	D11	I
49	NC	–
50	NC	–
51	D12	I
52	D13	I
53	D14	I
54	D15	I
55	VSS	V
56	ICP	V
57	DP0	B
58	DM0	B
59	DP1	B
60	DM1	B
61	VCC	V
62	VSS	V

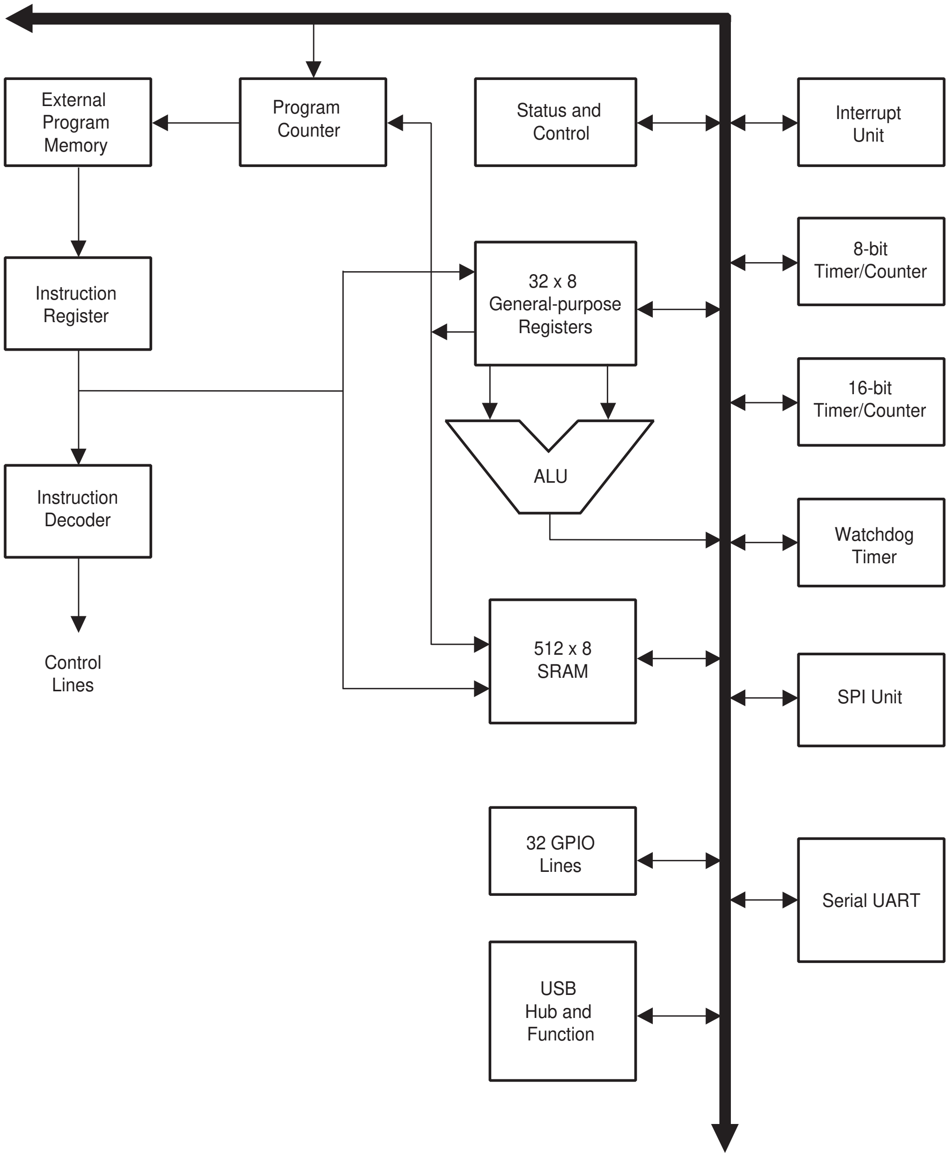
Pin Number	Signal	Type
63	CEXT2	O
64	DP2	B
65	DM2	B
66	DP3	B
67	DM3	B
68	DP4	B
69	DM4	B
70	PA0	B
71	PA1	B
72	PA2	B
73	PA3	B
74	PA4	B
75	VSS	V
76	NC	–
77	PA5	B
78	PA6	B
79	PA7	B
80	PB0	B
81	PB1	B

Pin Number	Signal	Type
82	PB2	B
83	PB3	B
84	VSS	V
85	PB4	B
86	PB5	B
87	PB6	B
88	PB7	B
89	PC0	B
90	PC1	B
91	PC2	B
92	PC3	B
93	PC4	B
94	PC5	B
95	PC6	B
96	PC7	B
97	PD0	B
98	PD1	B
99	VSS	V
100	NC	–

Signal Description

Name	Type	Function
V_{CC}	Power Supply/Ground	**5V Power Supply**
V_{SS}	Power Supply/Ground	**Ground**
CEXT1, 2	Power Supply/Ground	**External Capacitors for Power Supplies** – High quality 0.33 µF capacitors must be connected to CEXT1 and 2 for proper operation of the chip.
XTAL1	Input	**Oscillator Input** – Input to the inverting oscillator amplifier.
XTAL2	Output	**Oscillator Output** – Output of the inverting oscillator amplifier.
LFT	Input	**PLL Filter** – For proper operation of the PLL, this pin should be connected through a 0.01 µF capacitor in parallel with a 100Ω resistor in series with a 0.22 µF capacitor to ground (VSS). Both capacitors must be high quality ceramic.
DPO	Bi-directional	**Upstream Plus USB I/O** – This pin should be connected to CEXT1 through an external 1.5 kΩ.
DMO	Bi-directional	**Upstream Minus USB I/O**
DP[1:4]	Bi-directional	**Downstream Plus USB I/O** – Each of these pins should be connected to VSS through an external 15 kΩ resistor. DP[1:4] and DM[1:4] are the differential signal pin pairs to connect downstream USB devices.
DM[1:4]	Bi-directional	**Downstream Minus USB I/O** – Each of these pins should be connected to VSS through an external 15 kΩ resistor.
PA[0:7]	Bi-directional	**Port A[0:7]** – Bi-directional 8-bit I/O port with 4 mA drive strength.
PB[0:7]	Bi-directional	**Port B[0:7]** – Bi-directional 8-bit I/O port with 4 mA drive. PB[0,1,4:7] have dual functions as shown below: **Port Pin** **Alternate Function** PB0 T0, Timer/Counter0 External Input PB1 T1, Timer/Counter1 External Input PB4 SSN, SPI Slave Port Select or SCL, I2C Serial Bus Clock PB5 MOSI, SPI Slave Port Select Input PB6 MISO, SPI Master Data In, Slave Data Out PB7 SCK, SPI Master Clock Out, Slave Clock In
PC[0:7]	Bi-directional	**Port C[0:7]** – Bi-directional 8-bit I/O port with 4 mA drive strength.
PD[0:7]	Bi-directional	**Port D[0:7]** – Bi-directional I/O ports with 4 mA drive strength. PD[0:3,5] have dual functions as shown below: **Port Pin** **Alternate Function** PD0 RXD, Serial Input Port PD1 TXD, Serial Input Port PD2 INT0, External Interrupt 0 PD3 INT1, External Interrupt 1 PD5 OC1A Timer/Counter1 Output Compare A
TESTN	Input	**Test Pin** – This pin should be tied to ground.
SUSPEND	Output	**Suspend** – This pin is asserted when the AT43USB320A enters the Suspend status. In the 55618D, it is active low and in the 55618E and later versions, it is active high.

Figure 2. The AT43USB320A Enhanced RISC Architecture



Architectural Overview

The peripherals and features of the AT43USB320A microcontroller are similar to those of the AT90S8515, with the exception of the following modifications:

- External Program Memory
- No EEPROM
- No external data memory accesses
- No Analog Comparation
- Idle mode not supported
- USB Hub with attached function
- No internal pull-ups in the general-purpose I/O pin PA, PB, PC, PD

The embedded USB hardware of the AT43USB320A is a compound device, consisting of a 5 port hub with a permanently attached function on one port. The hub and attached function are two independent USB devices, each having its own device addresses and control endpoints. The hub has its dedicated interrupt endpoint, while the USB function has 2 additional programmable endpoints with separate 8-byte FIFOs.

The microcontroller always runs from a 12 MHz clock that is generated by the USB hardware. While the nominal and average period of this clock is 83.3 ns, it may have single cycles that deviate by ±20.8 ns during a phase adjustment by the SIE's clock/data separator of the USB hardware.

The microcontroller shares most of the control and status registers of the megaAVR™ Microcontroller Family. The registers for managing the USB operations are mapped into its SRAM space. The I/O section on page 16 summarizes the available I/O registers. The “AVR Register Set” on page 36 covers the AVR registers. Please refer to the Atmel AVR manual for more information.

The fast-access register file concept contains 32 x 8-bit general-purpose working registers with a single clock cycle access time. This means that during one single clock cycle, one Arithmetic Logic Unit (ALU) operation is executed. Two operands are output from the register file, the operation is executed, and the result is stored back in the register file – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing - enabling efficient address calculations. One of the three address pointers is also used as the address pointer for look-up tables in program memory. These added function registers are the 16-bit X-, Y- and Z-registers.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 2 on page 6 shows the AT43USB320A AVR Enhanced RISC microcontroller architecture.

In addition to the register operation, the conventional memory addressing modes can be used on the register file as well. This is enabled by the fact that the register file is assigned the 32 lowest Data Space addresses ($00 - $1 F), allowing them to be accessed as though they were ordinary memory locations.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the register file, $20 - $5F.

The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The program memory is executed with a single-level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is a downloadable SRAM or a mask programmed ROM.

With the relative jump and call instructions, the whole 24K address space is directly accessed. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the stack. The stack is effectively allocated in the general data SRAM, and consequently, the stack size is only limited by the total SRAM size and the usage of the SRAM. All user programs must initialize the Stack Pointer (SP) in the reset routine (before subroutines or interrupts are executed). The 10-bit SP is read/write accessible in the I/O space.

The 512-byte data SRAM can be easily accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps. A flexible interrupt module has its control registers in the I/O space with an additional global interrupt enable bit in the status register. All interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The interrupts have priority in accordance with their interrupt vector position. The lower the interrupt vector address, the higher the priority.

The General-purpose Register File

Table 1. AVR CPU General Purpose Working Register

Register	Address	Comment
R0	$00	
R1	$01	
R2	$02	
..		
R13	$0D	
R14	$0E	
R15	$0F	
R16	$10	
R17	$11	
..		
R26	$1A	X-register low byte
R27	$1B	X-register high byte
R28	$1C	Y-register low byte
R29	$1D	Y-register high byte
R30	$1E	Z-register low byte
R31	$1F	Z-register high byte

All register operating instructions in the instruction set have direct and single cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register, and the LDI instruction for load immediate constant data. These instructions apply to the second half of the registers in the register file – R16..R31. The general SBC, SUB, CP, AND, and OR and all other operations between two registers or on a single register apply to the entire register file.

As shown in Table 1, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-registers can be set to index any register in the file.

X-, Y- and Z-Registers

Registers R26..R31 contain some added functions to their general-purpose usage. These registers are address pointers for indirect addressing of the Data Space. The three indirect address registers X, Y, and Z are defined as:

X-register	15	XH			XL	0
	7		**0**	**7**		**0**
		R27 ($1B)			R26 ($1A)	

Y-register	15	YH			YL	0
	7		**0**	**7**		**0**
		R29 ($1D)			R28 ($1C)	

Z-register	15	ZH			ZL	0
	7		0	7		0
		R30 ($1F)			R31 ($1E)	

In the different addressing modes these address registers have functions as fixed displacement, automatic increment and decrement (see the descriptions for the different instructions).

ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all 32 general purpose working registers. Within a single clock cycle, ALU operations between registers in the register file are executed. The ALU operations are divided into three main categories – arithmetic, logical and bit-functions.

Program Memory

The AT43USB320A operates from an external program memory. Since all instructions are 16- or 32-bit words, the program memory is organized as X16. The AT43USB320A Program Counter (PC) is 16 bits wide, thus addressing the 64K program memory addresses.

Constant tables can be allocated within the entire program memory address space (see the LPM - Load Program Memory instruction description).

SRAM Data Memory

Table 3 summarizes how the AT43USB320A SRAM Memory is organized. The lower 608 Data Memory locations address the Register file, the I/O Memory and the internal data SRAM. The first 96 locations address the Register File + I/O Memory, and the next 512 locations address the internal data SRAM. The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement and Indirect with Post-increment. In the register file, registers R26 to R31 feature the indirect addressing pointer registers. Direct addressing reaches the entire data space.

The Indirect with Displacement mode features 63 address locations that reach from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented and incremented.

The 32 general purpose working registers, 64 I/O registers and the 1024 bytes of internal data SRAM in the AT43USB320A are all accessible through these addressing modes.

To manage the USB hardware, a special set of registers is assigned. These registers are mapped to SRAM space between addresses $1F00 and 1FFF. Table 3 and Table 4 give an overview of these registers.

Table 2. SRAM Organization

Register File		Data Address Space
R0		$0000
R1		$0001
R30		$001E
R31		$001F
I/O Registers		
$00		$0020
$01		$0021
$3E		$005E
$3F		$005F
		Internal SRAM
		$0060
		$0061
		$025E
		$045F
		USB Registers
		$1F00
		$1FFE
		$1FFF

Table 3. USB Hub and Function Registers

Address	Name	Function
$1FFD	FRM_NUM_H	Frame Number High Register
$1FFC	FRM_NUM_L	Frame Number Low Register
$1FFB	GLB_STATE	Global State Register
$1FFA	SPRSR	Suspend/Resume Register
$1FF9	SPRSIE	Suspend/Resume Interrupt Enable Register
$1FF7	UISR	USB Interrupt Status Register
$1FF5	UIAR	USB Interrupt Acknowledge Register
$1FF3	UIER	USB Interrupt Enable Register
$1FF2	UOVCER	Overcurrent Detect Register
$1FEF	HADDR	Hub Address Register
$1FEE	FADDR	Function Address Register
$1FE7	HENDP0_CNTR	Hub Endpoint 0 Control Register
$1FE5	FENDP0_CNTR	Function Endpoint 0 Control Register
$1FE4	FENDP1_CNTR	Function Endpoint 1 Control Register
$1FE3	FENDP2_CNTR	Function Endpoint 2 Control Register
$1FDF	HCSR0	Hub Controller Endpoint 0 Service Routine Register
$1FDD	FCSR0	Function Controller Endpoint 0 Service Routine Register
$1FDC	FCSR1	Function Controller Endpoint 1 Service Routine Register
$1FDB	FCSR2	Function Controller Endpoint 2 Service Routine Register
$1FD7	HDR0	Hub Endpoint 0 FIFO Data Register
$1FD5	FDR0	Function Endpoint 0 FIFO Data Register
$1FD4	FDR1	Function Endpoint 1 FIFO Data Register
$1FD3	FDR2	Function Endpoint 2 FIFO Data Register
$1FCF	HBYTE_CNT0	Hub Endpoint 0 Byte Count Register
$1FCD	FBYTE_CNT0	Function Endpoint 0 Byte Count Register
$1FCC	FBYTE_CNT1	Function Endpoint 1 Byte Count Register
$1FCB	FBYTE_CNT2	Function Endpoint 2 Byte Count Register
$1FC7	HSTR	Hub Status Register
$1FC5	HPCON	Hub Port Control Register
$1FBC	HPSTAT5	Hub Port 5 Status Register
$1FBB	HPSTAT4	Hub Port 4 Status Register
$1FBA	HPSTAT3	Hub Port 3 Status Register
$1FB9	HPSTAT2	Hub Port 2 Status Register
$1FB8	HPSTAT1	Hub Port 1 Status Register
$1FB4	HPSCR5	Hub Port 5 Status Change Register

Table 3. USB Hub and Function Registers (Continued)

Address	Name	Function
$1FB3	HPSCR4	Hub Port 4 Status Change Register
$1FB2	HPSCR3	Hub Port 3 Status Change Register
$1FB1	HPSCR2	Hub Port 2 Status Change Register
$1FB0	HPSCR1	Hub Port 1 Status Change Register
$1FAC	PSTATE5	Hub Port 5 Bus State Register
$1FAB	PSTATE4	Hub Port 4 Bus State Register
$1FAA	PSTATE3	Hub Port 3 Bus State Register
$1FA9	PSTATE2	Hub Port 2 Bus State Register
$1FA8	PSTATE1	Hub Port 1 Bus State Register
$1FA7	HCAR0	Hub Endpoint 0 Control and Acknowledge Register
$1FA5	FCAR0	Function Endpoint 0 Control and Acknowledge Register
$1FA4	FCAR1	Function Endpoint 1 Control and Acknowledge Register
$1FA3	FCAR2	Function Endpoint 2 Control and Acknowledge Register

Table 4. USB Hub and Function Registers

Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GLB_STATE	$1FFB	–			SUSP FLG	RESUME FLG	RMWUPE	CONFG	HADD EN
SPRSR	$1FFA	–	–	–	–	–	FRWUP	RSM	GLB SUSP
SPRSIE	$1FF9	–	–	–	–	–	FRWUP IE	RSM IE	GLB SUSP IE
UISR	$1FF7	SOF INT	EOF2 INT	–	FEP3 INT	HEP0 INT	FEP2 INT	FEP1 INT	FEP0 INT
UIAR	$1FF5	SOF INTACK	EOF2 INTACK	–	FEP3 INTACK	HEP0 INTACK	FEP2 INTACK	FEP1 INTACK	FEP0 INTACK
UIER	$1FF3	SOF IE	EOF2 IE	–	FEP3 IE	HEP0 IE	FEP2 IE	FEP1 IE	FEP0 IE
HADDR	$1FEF	SAEN	HADD6	HADD5	HADD4	HADD3	HADD2	HADD1	HADD0
FADDR	$1FEE	FEN	FADD6	FADD5	FADD4	FADD3	FADD2	FADD1	FADD0
HENDP0_CNTR	$1FE7	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP0_CNTR	$1FE5	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP1_CNTR	$1FE4	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
FENDP2_CNTR	$1FE3	EPEN	–	–	–	DTGLE	EPDIR	EPTYPE1	EPTYPE0
HCSR0	$1FDF	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX CEMPLETE
FCSR0	$1FDD	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
FCSR1	$1FDC	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
FCSR2	$1FDB	–	–	–	–	STALL SENT	RX SETUP	RX OUT PACKET	TX COMPLETE
HDR0	$1FD7	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR0	$1FD5	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR1	$1FD4	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
FDR2	$1FD3	DATA7	DATA6	DATA5	DATA4	DATA3	DATA2	DATA1	DATA0
HBYTE_CNT0	$1FCF	–	–	–	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT0	$1FCD	–	–	–	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT1	$1FCC	–	–	–	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
FBYTE_CNT2	$1FCB	–	–	–	BYTCT4	BYTCT3	BYTCT2	BYTCT1	BYTCT0
HSTR	$1FC7	–	–	–	–	OVLSC	LPSC	OVI	LPS
HPCON	$1FC5	–	HPCON2	HPCON1	HPCON0	–	HPADD2	HPADD1	HPADD0
HPSTAT5	$1FBC	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT4	$1FBB	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT3	$1FBA	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT2	$1FB9	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSTAT1	$1FB8	–	LSP	PPSTAT	PRSTAT	POCI	PSSTAT	PESTAT	PCSTAT
HPSCR5	$1FB4	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR4	$1FB3	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR3	$1FB2	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR2	$1FB1	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
HPSCR1	$1FB0	–	–	–	RSTSC	POCIC	PSSC	PESC	PCSC
PSTATE5	$1FAC	–	–	–	–	–	–	DPSTATE	DMSTATE
PSTATE4	$1FAB	–	–	–	–	–	–	DPSTATE	DMSTATE
PSTATE3	$1FAA	–	–	–	–	–	–	DPSTATE	DMSTATE
PSTATE2	$1FA9	–	–	–	–	–	–	DPSTATE	DMSTATE
PSTATE1	$1FA8	–	–	–	–	–	–	DPSTATE	DMSTATE
HCAR0	$1FA7	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK

Table 4. USB Hub and Function Registers (Continued)

Name	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
FCAR0	$1FA5	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK
FCAR1	$1FA4	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK
FCAR2	$1FA3	CTL DIR	DATA END	FORCE STALL	TX PACKET READY	STALL_SENT-ACK	RX_SETUP_ACK	RX_OUT_PACKET_ACK	TX_COMPLETE-ACK

I/O Memory

The I/O space definition of the AT43USB320A is shown in the following table:

Table 5. I/O Memory Space

I/O (SRAM) Address	Name	Function
$3F ($5F)	SREG	Status Register
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3B ($5B)	GIMSK	General Interrupt Mask Register
$3A ($5A)	GIFR	General Interrupt Flag Register
$39 ($59)	TIMSK	Timer/Counter Interrupt Mask Register
$38 ($58)	TIFR	Timer/Counter Interrupt Mask Register
$35 ($55)	MCUCR	MCU General Control Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8 bit)
$2F ($4F)	TCCR1A	Timer/Counter1 Control Register A
$2E ($4E)	TTCR1B	Timer/Counter0 Control Register B
$2D ($52)	TCNT1H	Timer/Counter1 High Byte
$2C ($52)	TCNT1L	Timer/Counter0 Low Byte
$2B ($4B)	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
$29 ($49)	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
$28 ($48)	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
$25 ($45)	ICR1H	T/C 1 Input Capture Register High Byte
$24 ($44)	ICR1L	T/C 1 Input Capture Register Low Byte
$21 ($41)	WDTCR	Watchdog Timer Counter Register
$1B ($4B)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$13($33)	PORTC	Data Register, Port C
$12 ($32)	PORTD	Data Register, Port D
$11 ($31)	DDRD	Data Direction Register, Port D
$10 ($30)	PIND	Input Pins, Port D
$0B (2B)	USR	UART Status Register
$0A (2A)	UCR	UART Control Register
$09 (29)	UBRR	UART Baud Rate Register

All AT43USB320A I/O and peripherals, except for the USB hardware registers, are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O registers within the address range $00 – $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the instruction set documentations of the AVR for more details. When using the I/O specific commands, IN and OUT, the I/O address $00 – $3F must be used. When addressing I/O registers as SRAM, $20 must be added to this address. All I/O register addresses throughout this document are shown with the SRAM address in parentheses.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

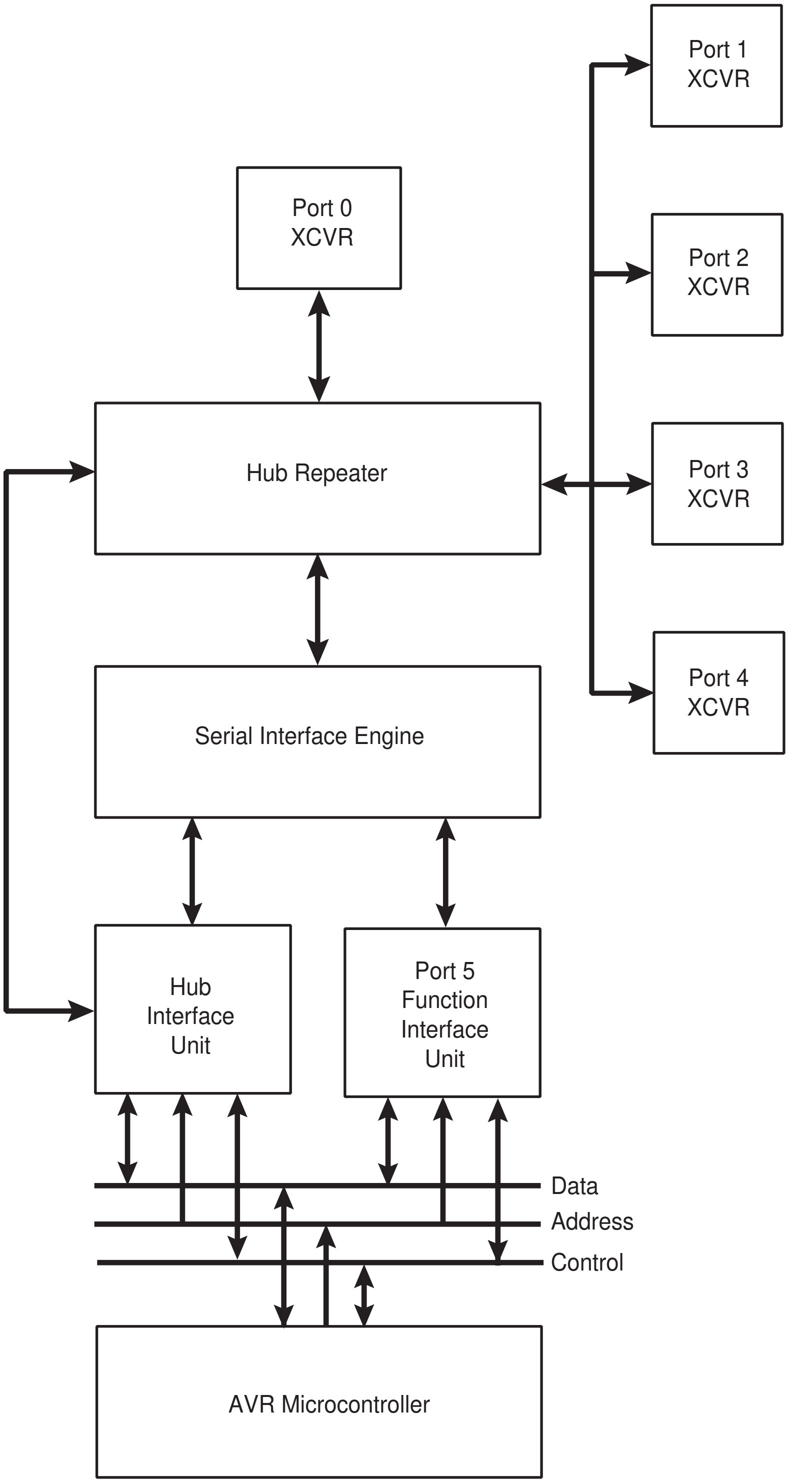
USB Hub

A block diagram of the USB hardware of the AT43USB320A is shown in Figure 3. The USB hub of the AT43USB320A has 5 downstream ports. The embedded function is permanently attached to Port 5. Ports 1 through 4 are available as external ports. The actual number of ports used is strictly defined by the firmware of the AT43USB320A and can vary from 0 to 4. Because the exact configuration is defined by firmware, ports 1 to 4 may even function as permanently attached ports as long as the Hub Descriptor identifies them as such.

USB Function

The embedded USB function has its own device address and has a default endpoint plus 2 other programmable endpoints with 8-byte FIFOs. Endpoints 1 - 3 can be programmed as interrupt IN or OUT or bulk IN or OUT endpoints.

Figure 3. USB Hardware



Functional Description

On-chip Power Supply

The AT43USB320A contains two on-chip power supplies that generate 3.3V with a capacity of 30 mA each from the 5V power input. The on-chip power supplies are intended to supply the AT43USB320A internal circuit and the 1.5K pull-up resistor only and should not be used for other purposes. External 0.33 µF filter capacitors are required at the power supply outputs, CEXT1 and 2. The internal power supplies can be disabled as described in the next paragraph.

The user should be careful when the GPIO pins are required to supply high-load currents. If the application requires that the GPIO supply currents beyond the capability of the on-chip power supply, the AT43USB320A should be supplied by an external 3.3V power supply. In this case, the 5V V_{CC} power supply pin should be left unconnected and the 3.3V power supplied to the chip through the CEXT1 and 2 pins.

I/O Pin Characteristics

The I/O pins of the AT43USB320A should not be directly connected to voltages less than V_{SS} or more than the voltage at the CEXT pins. If it is necessary to violate this rule, insert a series resistor between the I/O pin and the source of the external signal source that limits the current into the I/O pin to less than 2 mA. Under no circumstance should the external voltage exceed 5.5V. To do so will put the chip under excessive stress.

Oscillator and PLL

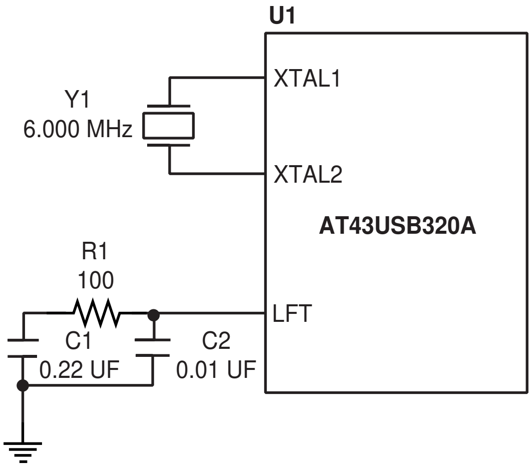
All clock signals required to operate the AT43USB320A are derived from an on-chip oscillator. To reduce EMI and power dissipation, the oscillator is designed to operate with a 6 MHz crystal. An on-chip PLL generates the high frequency for the clock/data separator of the Serial Interface Engine. In the suspended state, the oscillator circuitry is turned off.

The oscillator of the AT43USB320A is a special, low-drive type, designed to work with most crystals without any external components. The crystal must be of the parallel resonance type requiring a load capacitance of about 10 pF. If the crystal requires a higher value capacitance, external capacitors can be added to the two terminals of the crystal and ground to meet the required value. To assure quick start-up, a crystal with a high Q, or low ESR, should be used. To meet the USB hub frequency accuracy and stability requirements for hubs, the crystal should have an accuracy and stability of better than 100 PPM. The use of a ceramic resonator in place of the crystal is not recommended because a resonator would not have the necessary frequency accuracy and stability.

The clock can also be externally sourced. In this case, connect the clock source to the XTAL1 pin, while leaving XTAL2 pin floating. The switching level at the OSC1 pin can be as low as 0.47V and a CMOS device is required to drive this pin to maintain good noise margins at the low switching level.

For proper operation of the PLL, an external RC filter consisting of a series RC network of 100Ω and 0.22 µF in parallel with a 0.01 µF capacitor must be connected from the LFT pin to V_{SS}. Use only high-quality ceramic capacitors.

Figure 4. Oscillator and PLL



Reset and Interrupt Handling

The AT43USB320A provides 22 different interrupt sources with 13 separate reset vectors, each with a separate program vector in the program memory space. Eleven of the interrupt sources share 2 interrupt reset vectors. These 11 are the USB related interrupts. All interrupts are assigned individual enable bits which must be set (one) together with the I-bit in the status register in order to enable the interrupt.

The lowest addresses in the program memory space are automatically defined as the Reset and Interrupt vectors. The complete list of vectors is shown in Table 6. The list also determines the priority levels of the different interrupts. The lower the address, the higher is the priority level. RESET has the highest priority, and next is INT0 – the USB Suspend and Resume Interrupt, etc.

Table 6. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	External Reset, Power-on Reset and Watchdog Reset
2	$002	INT0	USB Suspend and Resume
3	$004	INT1	External Interrupt Request 1
4	$006	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$008	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$00A	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$00C	TIMER1, OVF	Timer/Counter1 Overflow
8	$00E	TIMER0, OVF	Timer/Counter0 Overflow
9	$010	SPI, STC	SPI Serial Transfer Complete
10	$012	UART RX	UART RX Complete
11	$014	UART UDRE	UART RX Data Receiver Output
12	$016	UART TX	UART TX Complete
13	$018	USB HW	USB Hardware

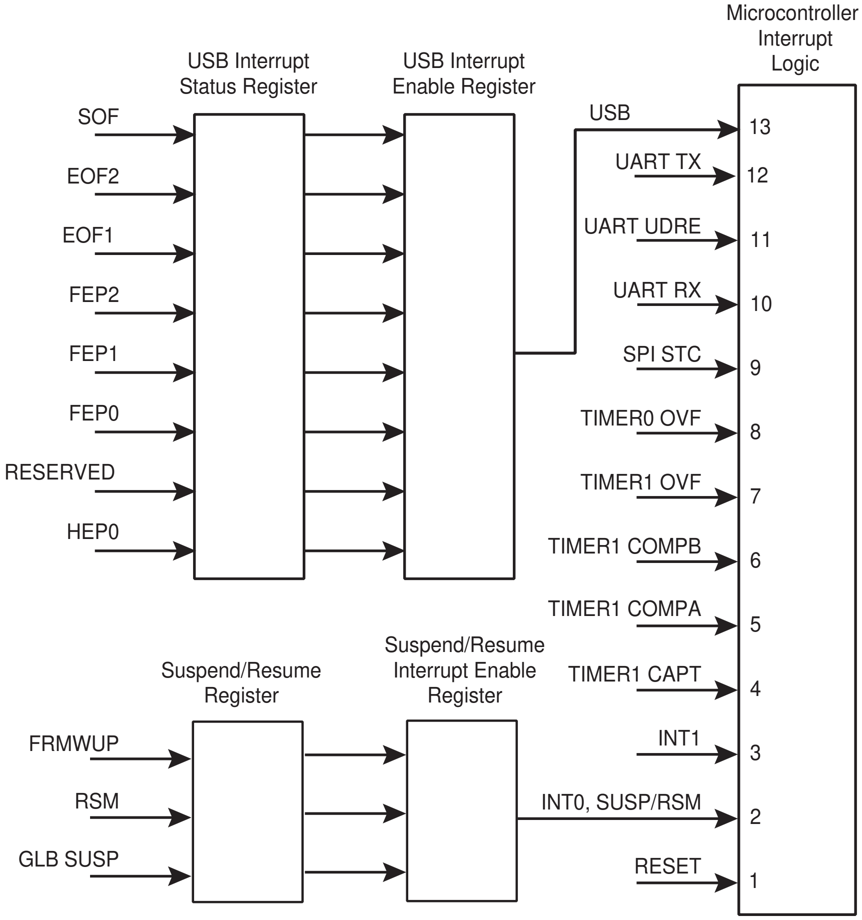
The most typical and general program setup for the Reset and Interrupt Vector Addresses are:

```
Address   Labels   Code                             Comments
$000               jmp                    RESET     ; Reset Handler
$004               jmp                    EXT_INT1  ; IRQ1 Handler
$00E               jmp                    TIM0_OVF  ; Timer0 Overflow
                                                    Handler
$018               jmp                    USB_HW    ; USB Handler
;
$00d      MAIN:    ldi r16, high (RAMEND)           ; Main Program
start
$00e               out SPH, r16
$00f               ldi r16, low (RAMEND)
$010               out SPL, r16
$011               <instr> xxx


...       ...      ...                    ...
```

USB related interrupt events are routed to reset vectors 13 and 2 through a separate set of interrupt, interrupt enable and interrupt mask registers that are mapped to the data SRAM space. These interrupts must be enabled though their control register bits. In the event an interrupt is generated, the source of the interrupt is identified by reading the interrupt registers. The USB frame and transaction related interrupt events, such as Start of Frame interrupt, are grouped in one set of registers: USB Interrupt Flag Register and USB Interrupt Enable Register. The USB Bus reset and suspend/resume are grouped in another set of registers: Suspend/Resume Register and Suspend/Resume Interrupt Enable Register.

Figure 5. AT43USB320A Interrupt Structure



Reset Sources

The AT43USB320A has four sources of reset:

- **Power-on Reset** – The MCU is reset when the supply voltage is below the power-on reset threshold.
- **External Reset** – The MCU is reset when a low level is present on the RESET pin for more than 50 ns.
- **Watchdog Reset** – The MCU is reset when the watchdog timer period expires and the watchdog is enabled.
- **USB Reset** – A USB bus reset is defined as a SE0 (single ended zero) of at least 4 slow speed USB clock cycles received by Port0. The internal reset pulse to the USB hardware and microcontroller lasts for 24 oscillator periods.

When the USB hardware is reset, the compound device is de-configured and has to be re-enumerated by the host. When the microcontroller is reset, all I/O registers are then set to their initial values, and the program starts execution from address $000. The instruction placed in address $000 must be a JMP instruction to the reset handling routine. If the program never enables an interrupt source, the interrupt vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 6 shows the reset logic. The user can select the start-up time according to typical oscillator start-up. The number of WDT oscillator cycles used for each time-out is shown in Table 7.

Figure 6. Reset Logic

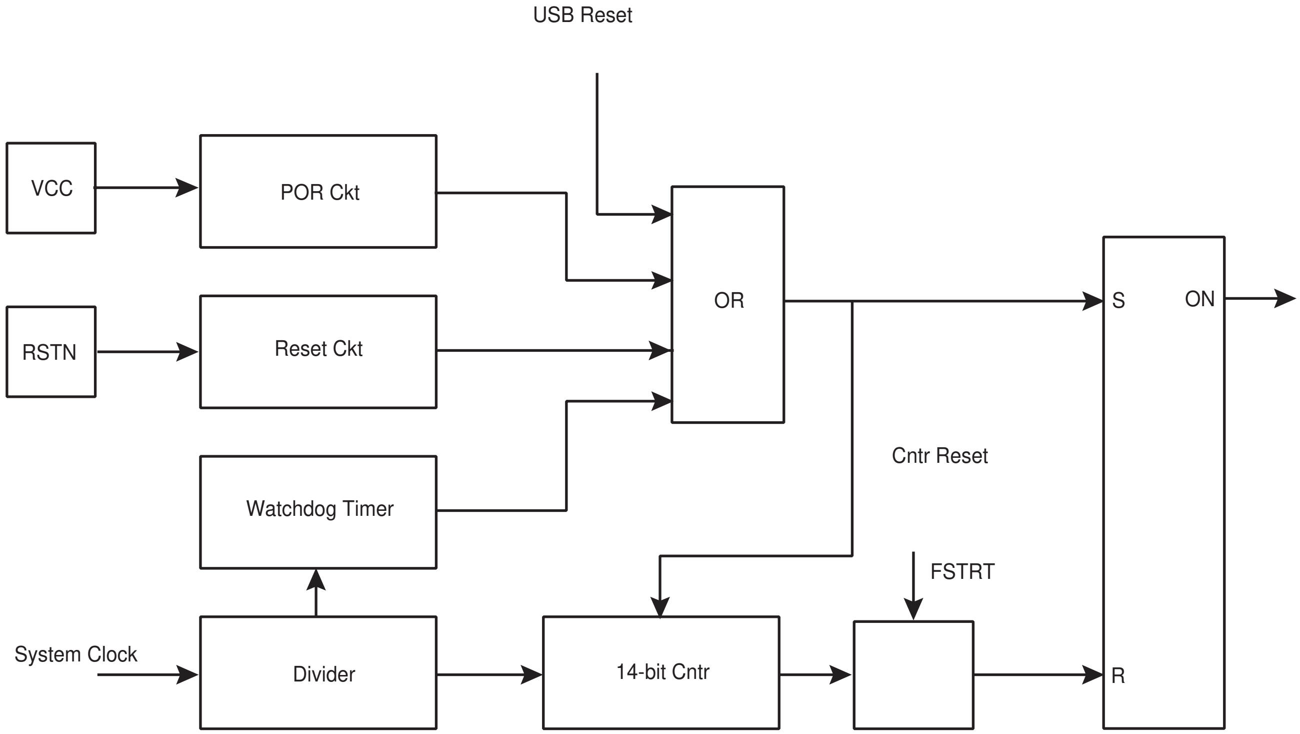


Table 7. Number of Watchdog Oscillator Cycles

FSTRT	Time-out at V_{CC} = 5V	Number of WDT cycles
Programmed	1.1 ms	1K
Unprogrammed	16.0 ms	16K

Power-on Reset

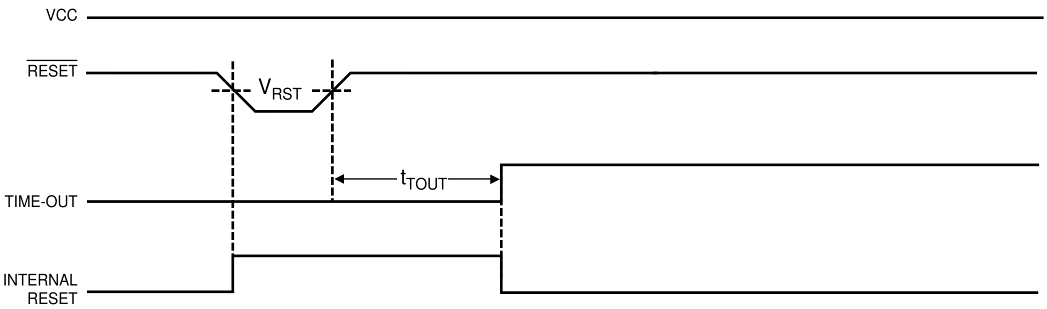
A Power-on Reset (POR) circuit ensures that the device is reset from power-on. An internal timer clocked from the Watchdog timer oscillator prevents the MCU from starting until after a certain period after V_{CC} has reached the power-on threshold voltage, regardless of the V_{CC} rise time.

If the build-in start-up delay is sufficient, RESET can be connected to V_{CC} directly or via an external pull-up resistor. By holding the pin low for a period after V_{CC} has been applied, the Power-on Reset period can be extended.

External Reset

An external reset is generated by a low-level on the RESET pin. Reset pulses longer than 200 ns will generate a reset. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage - V_{RST} on its positive edge, the delay timer starts the MCU after the Time-out period t_{TOUT} has expired.

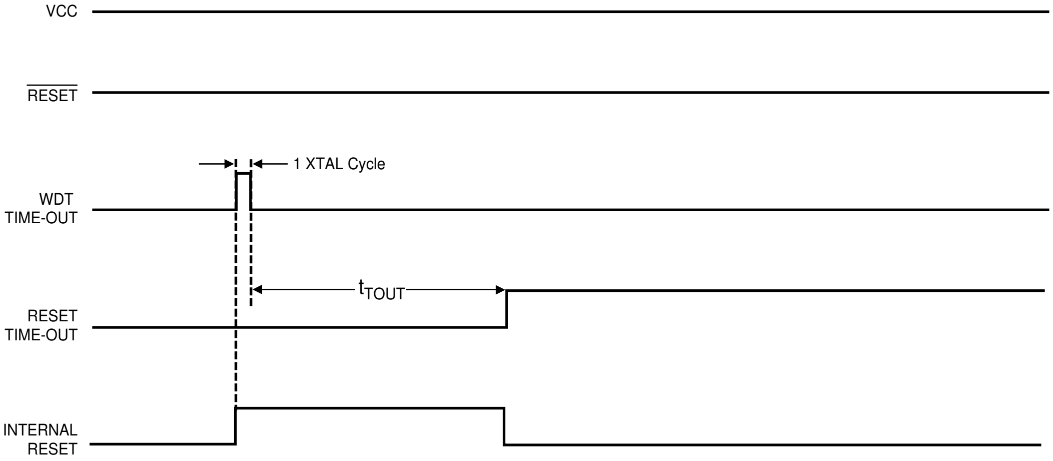
Figure 7. External Reset During Operation



Watchdog Timer Reset

When the watchdog times out, it will generate a short reset pulse of 1 XTAL cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}.

Figure 8. Watchdog Reset During Operation



Non-USB Related Interrupt Handling

The AT43USB320A has two non-USB 8-bit Interrupt Mask control registers; GIMSK (General Interrupt Mask Register) and TIMSK (Timer/Counter Interrupt Mask Register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set (one) the I-bit to enable nested interrupts. The I-bit is set (one) when a Return from Interrupt instruction, RETI, is executed.

For Interrupts triggered by events that can remain static (e.g. the Output Compare register1 matching the value of Timer/Counter1) the interrupt flag is set when the event occurs. If the interrupt flag is cleared and the interrupt condition persists, the flag will not be set until the event occurs the next time.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hard-ware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is active.