An ISO 9000 Registered Company

Atmel

AVR™

Atmel Corporation
Enhanced RISC Microcontrollers
Data Book

May 1996

Preliminary

ATMEL®

2325 Orchard Parkway, San Jose, CA 95131

Important Notice

Atmel guarantees that its circuits will be free from defects of material and workmanship under normal use and service, and that these circuits will perform to current specifications in accordance with, and subject to, the Company's standard warranty which is detailed in Atmel's Purchasing Order Acknowledgment.

Atmel reserves the right to change devices or specifications detailed in this data book at any time without notice, and assumes no responsibility for any errors within this document. Atmel does not make any commitment to update this information. Atmel assumes no responsibility for the use of any circuits described in this data book, nor does the Company assume responsibility for the functioning of undescribed features or parameters.

In the absence of a written agreement to the contrary, Atmel assumes no liability with respect to the use of semiconductor devices described in this data book for applications assistance, customers' product design or infringement of patents or copyrights of third parties.

Atmel's products are not authorized for use as critical components in life support devices or systems and the use as such implies that user bears all risk of such use.

If Atmel is an approved vendor on a Standard Microcircuit Drawing (SMD), the Atmel similar part number specification is compliant with the SMD.

Printed on recycled paper.

Atmel Corporation designs, manufactures, and markets high quality and high performance CMOS memory, logic and analog integrated circuits. Founded in 1984, the Company serves the manufacturers of computation, communications and instrumentation equipment in commercial, industrial and military environments.

Atmel's broad line of products provide customers with a variety of solutions to their memory and logic applications. Atmel offers high-density, high-speed memory and logic standard products as well as custom gate arrays.

Atmel guarantees quality and reliability by fabricating all products— no matter what their intended application— to meet or exceed the specifications of Military Standard 883.

Whether you are new to programmable logic or an experienced user, Atmel is committed to your success. If you have any questions or would like to place an order, please contact your local Atmel sales office as listed in the back of this data book, or contact Atmel's corporate headquarters:

Atmel Corporation
2325 Orchard Parkway
San Jose, CA 95131
TEL: (408) 441-0311
FAX: (408) 436-4300

Fax-on-Demand
North America:
1-(800) 29-ATMEL
1-(800) 292-8635
International:
1-(408) 441-0732
e-mail
literature@atmel.com
Web Site
http://www.atmel.com
BBS
1-(408) 436-4309

We thank you for considering Atmel semiconductors.

Table of Contents

(continued)

Section 3 AT90S2312 (Continued)

Section 4 AT90S8414

Section 5 Instruction Set

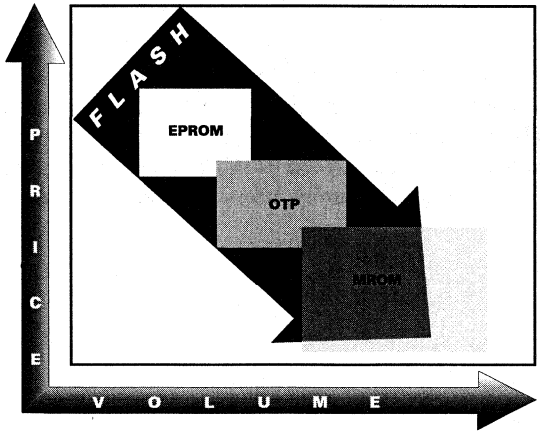
Table of Contents

AVR™ Enhanced RISC Microcontrollers

Atmel Corporation is a leading manufacturer of a broad range of high performance, low power nonvolatile memory and logic integrated circuits (ICs) that focus on the telecommunications, computer, networking, consumer and automotive markets. Atmel's Flash-based microcontroller families incorporate advanced single voltage Flash memory technology in the industry's broadest line of Flash and EEPROM based products.

With the Flash memory-based microcontrollers from Atmel, you can achieve safe, easy reconfigurability:

-Change code in seconds, shortening the development cycle
-Stock just one part
-Zero scrap due to misprogramming
-Accelerate product testing
-Make changes remotely
-Customize each product on the line



The *AVR* enhanced RISC microcontrollers are based on a new RISC architecture that has been developed to take advantage of semiconductor integration and software capabilities in the 1990's. The resulting microcontrollers offer the highest MIPS/milliwatt capability available in the 8 bit MCU market.

High level languages are rapidly becoming the standard programming methodology for embedded microcontrollers due to improved time-to-market and simplified maintenance support. The *AVR* architecture was developed in conjunction with C language experts to ensure that the hardware and software work hand in hand to develop highly efficient, high performance code.

In order to optimize code size, performance and power consumption, the ***AVR*** architecture has incorporated a large fast-access register file and fast single-cycle instructions.

The fast-access RISC register file consists of 32 general purpose working resisters. Traditional accumulator based architectures require large amounts of program code for data transfers between the accumulator and memory. With 32 working registers (accumulators) in the ***AVR*** these data transfers are eliminated.

The ***AVR*** pre-fetches an instruction during the previous instruction execution and then executes in a single cycle. In other CISC- and RISC-like architectures, the external oscillator clock is divided down (by as much as 12 times) to the traditional internal execution cycle. The ***AVR*** enhanced RISC microcontrollers execute an instruction in a single clock cycle and are the first true RISC machines in the 8 bit market.

The ***AVR*** architecture supports a complete spectrum of price performance from simple small pin count controllers to high range devices with large on-chip memories. The Harvard style architecture directly addresses up to 8M bytes of program memory and 8M bytes of data memory. The register file is dual mapped and can be addressed as part of the on-chip SRAM memory to enable fast context switching.

The ***AVR*** enhanced RISC microcontroller family is manufactured with Atmel's low-power nonvolatile CMOS technology. The on-chip In-System-Programmable (ISP) downloadable Flash memory allows the program memory to be reprogrammed in system through an SPI serial port or by a conventional memory programmer. By combining an enhanced RISC architecture with downloadable Flash memory on the same chip, the ***AVR*** family offers a powerful solution to embedded control applications.

Contents

8-Bit AVR Microcontroller with 1K bytes Downloadable Flash

Features

- Utilizes the AVR™ Enhanced RISC Architecture
- AVR™ - High Performance and Low Power RISC Architecture
- 83 Powerful Instructions - Most Single Clock Cycle Execution
- 1K bytes of In-System Reprogrammable Downloadable Flash
  SPI Serial Interface for Program Downloading
  Endurance: 1,000 Write/Erase Cycles
- 128 bytes EEPROM
  Endurance: 100,000 Write/Erase Cycles
- 32 x 8 General Purpose Working Registers
- 15 Programmable I/O Lines
- VCC Min.: 2.7 V
- Fully Static Operation
- One 8-Bit Timer/Counter with Separate Prescaler
- External and Internal Interrupt Sources
- Programmable Watchdog Timer
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Programming Lock for Software Security
- 20-Pins Device

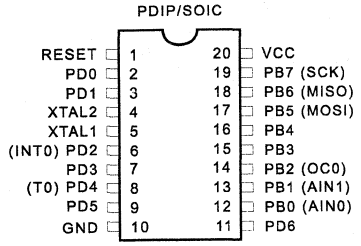
Description

The AT90S1300 is a low-power CMOS 8 bit microcontroller based on the *AVR* enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S1300 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

The *AVR* core is based on an enhanced RISC architecture that combines a rich instruction set with the 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (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.

(continued)

Pin Configuration



Block Diagram

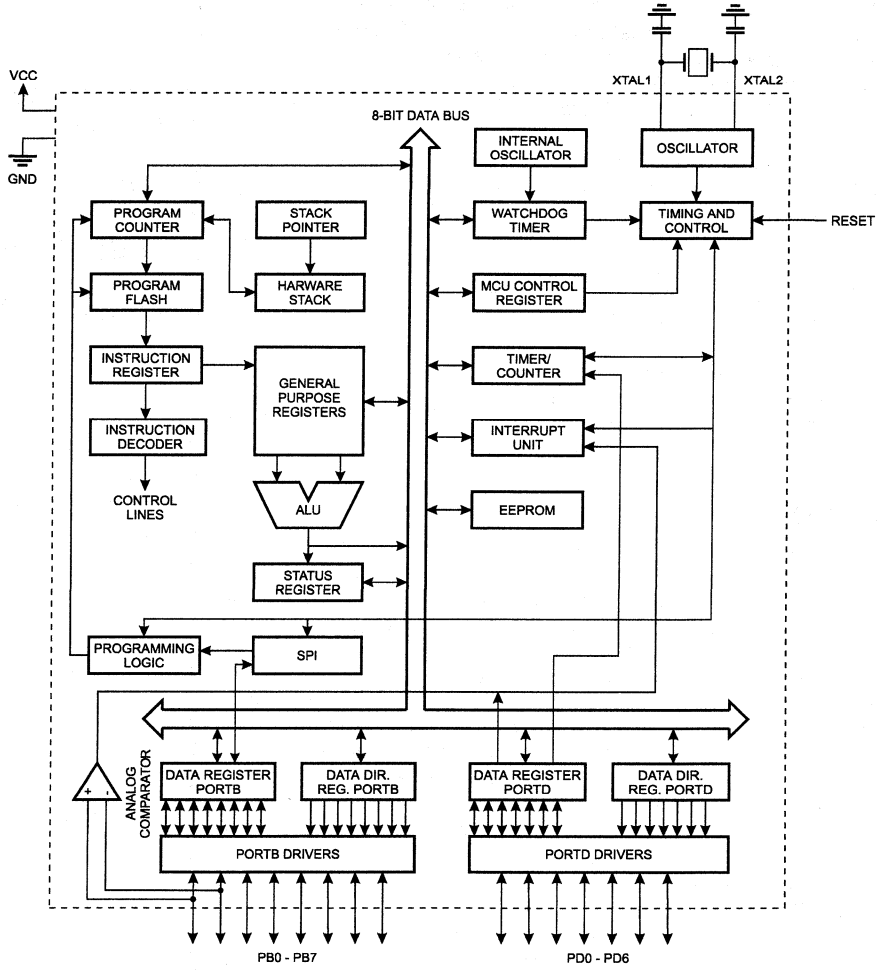


Figure 1: The AT90S1300 Block Diagram

Description (Continued)

The architecture supports high level languages efficiently as well as extremely dense assembler code programs. The AT90S1300 provides the following features: 1K bytes of Downloadable Flash, 128 bytes EEPROM, 15 general purpose I/O lines, 32 general purpose working registers, flexible timer/counter with compare modes, internal and external interrupts, programmable Watchdog Timer with internal oscillator, an SPI serial port for program downloading and two software selectable power saving modes. The Idle Mode stops the CPU while allowing the registers, timer/counter and interrupt system to continue functioning. The power down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8 bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT90S1300 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90S1300 *AVR* is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC
Supply voltage pin.

GND
Ground pin.

Port B (PB7..PB0)
Port B is an 8-bit bi-directional I/O port. Port pins can provide internal pullups (selected for each bit). PB0 and PB1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip analog comparator. The Port B output buffers can sink 20mA and can drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

Port B also serves the functions of various special features of the AT90S1300 as listed on Page 2-24.

Port D (PD6..PD0)
Port D has seven bi-directional I/O pins with internal pullups, PD6..PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Port D also serves the functions of various special features of the AT90S1300 as listed on Page 2-24.

RESET
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier

Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively , of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

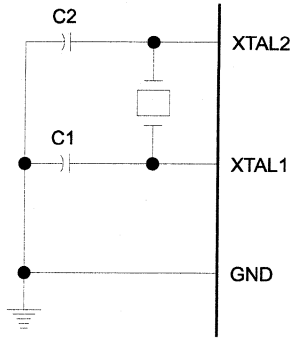


Figure 2: Oscillator Connections

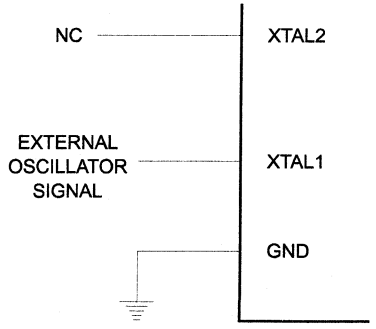


Figure 3: External Clock Drive Configuration

AT90S1300 *AVR* Enhanced RISC Microcontroller CPU

The AT90S1300 *AVR* RISC microcontroller is upward compatible with the *AVR* Enhanced RISC Architecture. The programs written for the AT90S1300 MCU are compatible with the range of *AVR* 8-bit MCUs (AT90Sxxxx) with respect to source code and clock cycles for execution.

Architectural Overview

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 ALU (Arithmetic Logic Unit) 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.

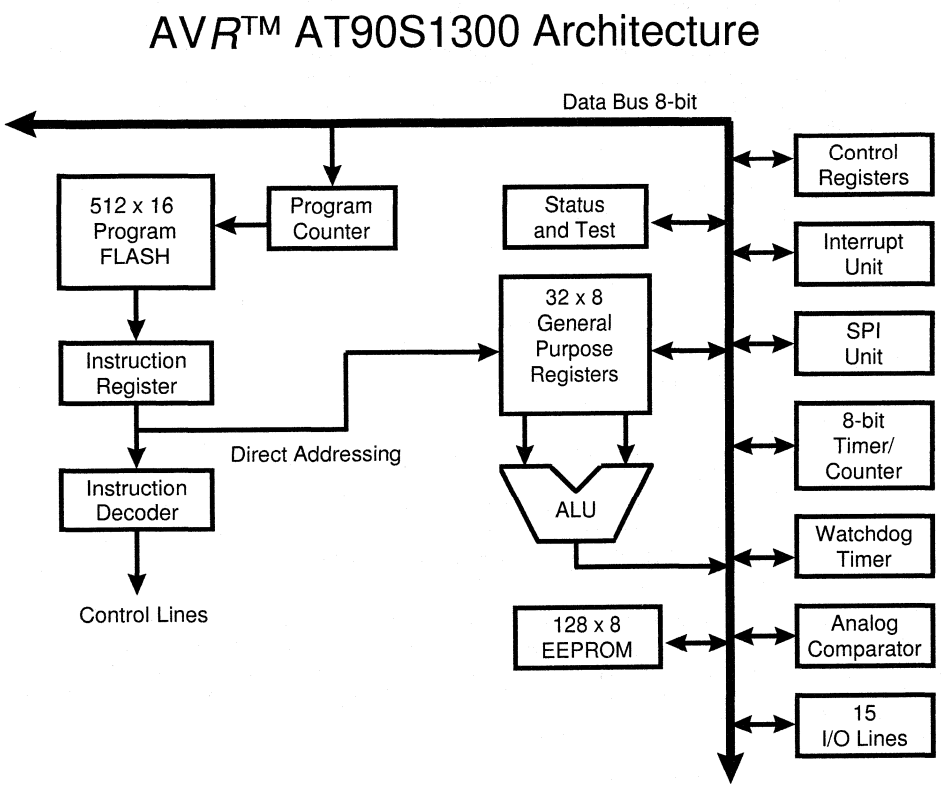


Figure 4: The AT90S1300 *AVR* Enhanced RISC Architecture

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S1300 *AVR* Enhanced RISC microcontroller architecture. The *AVR* uses a Harvard architecture concept - with separate memories and buses for program and data

memories. The program memory is accessed 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 in-system downloadable Flash memory.

With the relative jump and relative call instructions, the whole 512 address space is directly accessed. All *AVR* instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address program counter (PC) is stored on the stack. The stack is a 4 level deep hardware stack dedicated for subroutines and interrupts.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, A/D-converters, and other I/O functions. 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 the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt address vector the higher priority.

The General Purpose Register File

Figure 5 shows the structure of the 32 general purpose registers in the CPU.

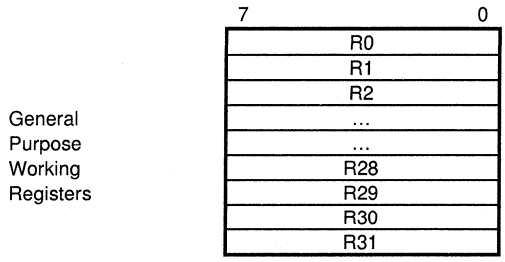


Figure 5: *AVR* CPU General Purpose Working Registers

All the 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, 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, OR and all other operations between two registers or on a single register apply to the entire register file.

The ALU - Arithmetic Logic Unit

The high-performance *AVR* ALU operates in direct connection with all the 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, logic and bit-functions. Some microcontrollers in the *AVR* product family feature a hardware multiplier in the arithmetic part of the ALU.

The Downloadable Flash Program Memory

The AT90S1300 contains 1K bytes on-chip downloadable Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 512 x 16 words. The Flash memory has an endurance of at least 1000 write/erase cycles.

The AT90S1300 Program Counter is 9 bit wide, thus addressing the 512 words Flash program memory.

See Page 2-32 for a detailed description on Flash data downloading.

The Program and Data Addressing Modes

The AT90S1300 *AVR* Enhanced RISC Microcontroller supports powerful and efficient addressing modes. This section describes the different addressing modes supported in the AT90S1300. In the figures, OP means the operation code part of the instruction word.

REGISTER DIRECT, SINGLE REGISTER Rd

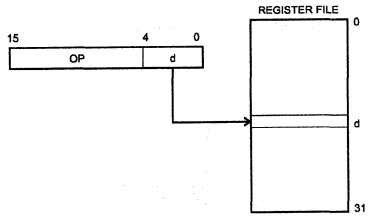


Figure 6: Direct Single Register Addressing

The operand is contained in register d (Rd).

REGISTER DIRECT, TWO REGISTERS Rd AND Rr

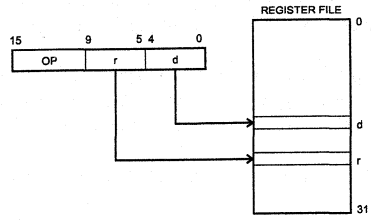


Figure 7: Direct Register Addressing, Two Registers

Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O DIRECT

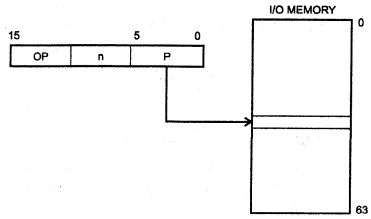


Figure 8: I/O Direct Addressing

Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

RELATIVE PROGRAM ADDRESSING, RJMP AND RCALL

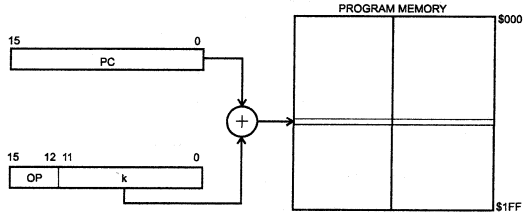


Figure 9: Relative Program Memory Addressing

Program execution continues at address PC + k. The relative address k is in the range from -2K to +(2K - 1).

Subroutine and Interrupt Hardware Stack

The AT90S1300 uses a 4 level deep hardware stack for subroutines and interrupts. The hardware stack is 9 bit wide, and stores the Program Counter - PC - return address while subroutines and interrupts are executed.

RCALL instructions and interrupts push the PC return address onto stack level 0, and the data in the other stack levels 1-3 are pushed one level deeper in the stack. When a RET or RETI instruction is executed the returning PC is fetched from the stack level 0, and the data in the other stack levels 1-3 are popped one level in the stack.

If more than 4 subsequent subroutine calls or interrupts are executed, only the most recent 4 return addresses are stored in the stack.

The EEPROM Data Memory

The AT90S1300 contains 128 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on Page 2-21 specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register. For the SPI data downloading, see Page 2-36 for a detailed description.

Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The *AVR* CPU is driven by the System Clock Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 10 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

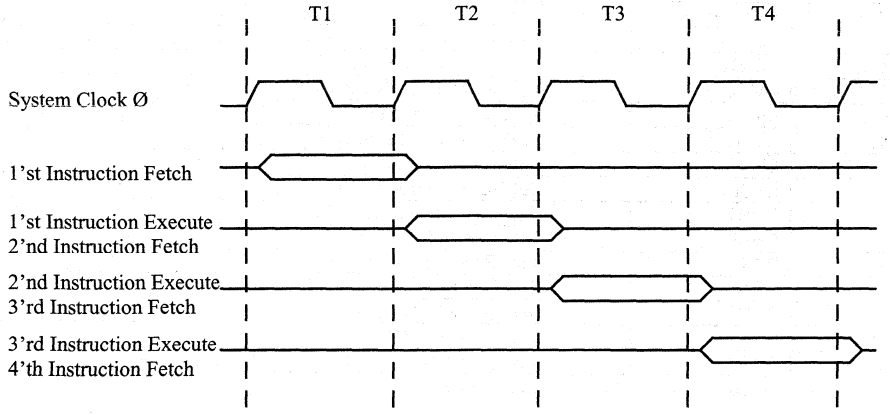


Figure 10: The Parallel Instruction Fetches and Instruction Executions

Figure 11 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

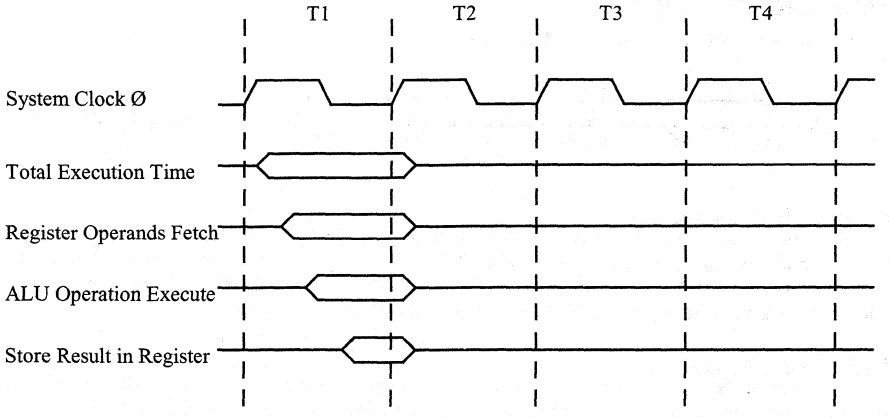


Figure 11: Single Cycle ALU Operation

I/O Memory

The I/O space definition of the AT90S1300 is shown in the following table:

Table 1: The AT90S1300 I/O Space

Address Hex	Name	Function
$3F	SREG	Status REGister
$3B	GIMSK	General Interrupt MaSK register
$39	TIMSK	Timer/Counter Interrupt MaSK register
$38	TIFR	Timer/Counter Interrupt Flag register
$35	MCUCR	MCU general Control Register
$33	TCCR0	Timer/Counter 0 Control Register
$32	TCNT0	Timer/Counter 0 (8-bit)
$31	OCR0	Output Compare Register 0
$21	WDTCR	Watchdog Timer Control Register
$1E	EEAR	EEPROM Address Register
$1D	EEDR	EEPROM Data Register
$1C	EECR	EEPROM Control Register
$18	PORTB	Data Register, Port B
$17	DDRB	Data Direction Register, Port B
$16	PINB	Input Pins, Port B
$12	PORTD	Data Register, Port D
$11	DDRD	Data Direction Register, Port D
$10	PIND	Input Pins, Port D
$08	ACSR	Analog Comparator Control and Status Register

Note: reserved and unused locations are not shown in the table

All the different AT90S1300 I/Os and peripherals are placed in the I/O space. The different 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.
The different I/O and peripherals control registers are explained in the following sections.

THE STATUS REGISTER - SREG

The AVR status register - SREG - at I/O space location $3F is defined as:

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

Bit 7 - I : Global Interrupt Enable:
The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in the interrupt mask registers - GIMSK/TIMSK. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the GIMSK/TIMSK values. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

Bit 6 - T : Bit Copy Storage:
The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

Bit 5 - H : Half Carry Flag:
The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

Bit 4 - S : Sign Bit, S = N ⊕ V :
The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

Bit 3 - V : Two's Complement Overflow Flag:
The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

Bit 2 - N : Negative Flag:
The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 1 - Z : Zero Flag:
The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 0 - C : Carry Flag:
The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information.

Reset and Interrupt Handling

The AT90S1300 provides 4 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All the 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 2: Reset and Interrupt Vectors. 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 External Interrupt Request 0 etc.

Table 2: Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	Hardware Pin and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	TIMER0, COMP0	Timer/Counter0 Compare Match
4	$003	TIMER0, OVF0	Timer/Counter0 Overflow
5	$004	ANA_COMP	Analog Comparator

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

```
Address     Labels      Code                Comments
$000                    rjmp  RESET         ; Reset handle
$001                    rjmp  EXT_INT0      ; IRQ0 handle
$002                    rjmp  TIM0_COMP     ; Timer0 comp. Handle
$003                    rjmp  TIM0_OVF      ; Timer0 overflow handle
$004                    rjmp  ANA_COMP      ; Analog Comparator Handle
;
$005        MAIN:       <instr>  xxx        ; Main program start
  ...         ...          ...     ...
```

RESET

A Reset State is enabled by a high level on the RESET pin. The pin must be held high for at least two crystal clock cycles. All internal 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 RJMP - relative jump - 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.

INTERRUPT HANDLING

The AT90S1300 has two Interrupt Mask control registers GIMSK - General Interrupt MaSK register - at I/O space address $3B and the TIMSK - Timer/Counter Interrupt MaSK register at I/O address $39.

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 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 register0 matching the value of Timer/Counter0) 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, hardware 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.

THE GENERAL INTERRUPT MASK REGISTER - GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B	-	INT0	-	-	-	-	-	-	GIMSK
Read/Write	R	R/W	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - Res : Reserved bit:
This bit is a reserved bit in the AT90S1300 and always read zero.

Bit 6 - INT0 : External Interrupt Request 0 Enable:
When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bit 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. If the INT0 pin for external interrupts shall be activated, the DDD2 bit in the Data Direction Register PORTD (DDRD) must be cleared (zero) to force an input pin. See also "External Interrupts".

Bits 5..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read as zero.

THE TIMER/COUNTER INTERRUPT MASK REGISTER - TIMSK

Bit	7	6	5	4	3	2	1	0	
$39	-	-	-	-	-	-	TOIE0	OCIE0	TIMSK
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..2 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

Bit 1 - TOIE0 : Timer/Counter0 Overflow Interrupt Enable:
When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $003) is executed if an overflow in Timer/Counter0 occurs. The Overflow Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 0 - OCIE0 :Timer/Counter0 Output Compare Match Interrupt Enable:
When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at vector $002) is executed if a compare match in Timer/Counter0 occurs. The Compare Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

THE TIMER/COUNTER INTERRUPT FLAG REGISTER - TIFR

Bit	7	6	5	4	3	2	1	0	
$38	-	-	-	-	-	-	TOV0	OCF0	TIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..2 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

Bit 1 - TOV0 : Timer/Counter0 Overflow Flag:
The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

Bit 0 - OCF0 : Output Compare Flag 0:
The OCF0 bit is set (one) when a compare match occurs between the Timer/Counter0 and the compared data in OCR0 - Output Compare Register0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0 is cleared by writing a logic one to the flag.. When the SREG I-bit, and OCIE0 (Timer/Counter0 Compare match Interrupt Enable), and OCF0 are set (one), the Timer/Counter0 Compare match Interrupt is executed.

EXTERNAL INTERRUPTS

The external interrupts are triggered by the INT0 pin. Since this pin is an alternate function pin in the general I/O ports, the corresponding pin must be set as an input pin in the data direction register - DDRX.

The external interrupt is set up as indicated in the specification for the general interrupt mask register - GIMSK.

INTERRUPT RESPONSE TIME

The interrupt execution response times for all the enabled AVR™ interrupts are 4 clock cycles. After the 4 clock cycles the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (9 bits) is pushed onto the Stack.

A return from an interrupt handling routine takes 4 clock cycles. During these 4 clock cycles, the Program Counter (9 bits) is popped back from the Stack.

Note that the Status Register - SREG - is not handled by the AVR™ hardware, neither for interrupts nor for subroutines. For the routines requiring a storage of the SREG, this must be performed by user software.

Note that the Subroutine and Interrupt Stack is a 4 level true hardware stack, and if more than 4 nested subroutines and interrupts are executed, only the most recent 4 return addresses are stored.

THE MCU CONTROL REGISTER - MCUCR

The MCU Control Register contains general microcontroller control bits for general MCU control functions.

Bit	7	6	5	4	3	2	1	0	
$35	-	-	**SE**	**SM**	-	-	**ISC01**	**ISC00**	**MCUCR**
Read/Write	R	R	R/W	R/W	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7, 6 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

Bit 5 - SE : Sleep Enable:
The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

Bit 4 - SM : Sleep Mode:
This bit selects between the two available sleep modes - the Idle Mode and the Power Down Mode.

To enter the sleep modes, the SE bit must be set (one) and a SLEEP instruction must be executed. The instruction following SLEEP is executed before entering sleep mode. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine and resumes execution from the address following the last executed instruction. If reset occurs while the MCU is in Sleep Mode, the MCU awakes and executes from the reset vector.

When the SM bit is cleared (zero), the SLEEP instruction forces the MCU into the Idle Mode stopping the CPU but allowing the General Purpose Working Registers, Timer/Counters, SPI port, and the interrupt system to continue operating.

When the SM bit is set (one), the SLEEP instruction forces the MCU into the Power Down Mode saving the General Purpose Working Registers, but freezing the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

Bits 3, 2 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

Bits 1, 0 - ISC01, ISC00 : Interrupt Sense Control 0 bit 1 and bit 0:
The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask in the GIMSK register is set. The level and edges on the external INT0 pin that activate the interrupt are defined as:

Table 3: Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Note: When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Timer / Counter

The AT90S1300 provides one general purpose 8-bit Timer/Counter. The Timer/Counter has prescaling selection from the 10-bit prescaling timer. The Timer/Counter can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

The Timer/Counter Prescaler

Figure 12 shows the general Timer/Counter prescaler.

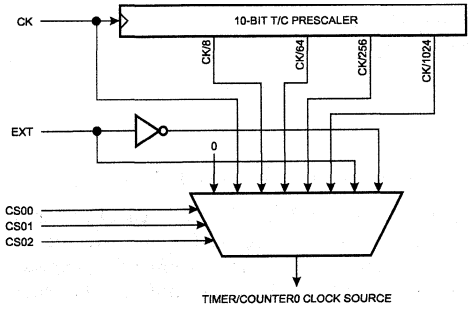


Figure 12: Timer/Counter0 Prescaler

The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK is the oscillator clock. For the Timer/Counter, added selections as CK, external source and stop, can be selected as clock sources.

The 8-bit Timer/Counter0

Figure 13 shows the block diagram for Timer/Counter0.

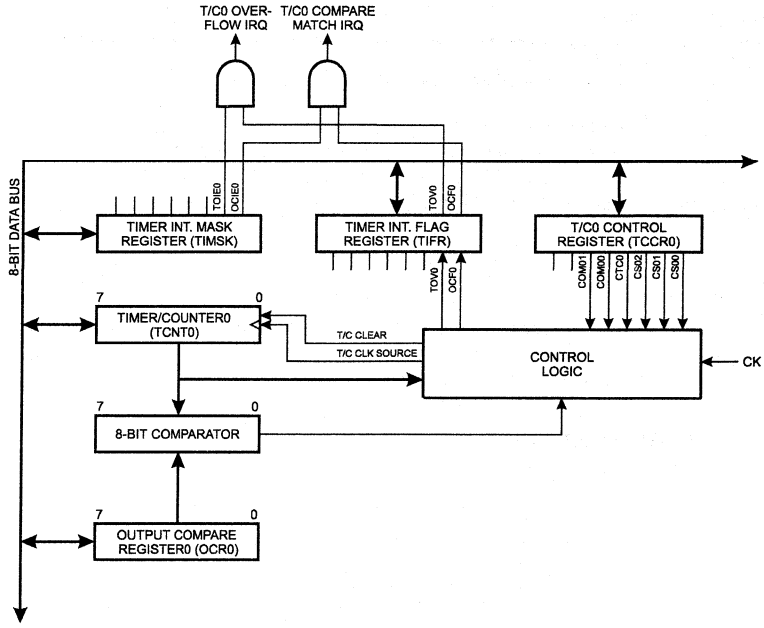


Figure 13: Timer/Counter 0 Block Diagram

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter0 Control Register - TCCR0. The different status flags (overflow and compare match) are found in the Timer/Counter Interrupt Flag Register - TIFR. Control signals are found in the Timer/Counter0 Control Register - TCCR0. The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time for the external clock being low and high must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter0 supports an Output Compare function using the Output Compare Register 0 - OCR0 as the data source to be compared to the Timer/Counter0 contents. The Output Compare functions include optional clearing of the counter on compare matches, and actions on the Output Compare pin 0 on compare matches. The Output Compare pin 0 function makes the Timer/Counter0 useful for PWM (Pulse Width Modulation) functions.

THE TIMER/COUNTER0 CONTROL REGISTER - TCCR0

Bit	7	6	5	4	3	2	1	0	
$33	-	-	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7,6 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

Bits 5,4 - COM01, COM00 : Compare Output Mode0, bit 1 and 0:
The COM01 and COM00 control bits determine any output pin action following a compare match in Timer/Counter0. Any output pin actions are effective on pin OC0 - Output Compare pin 0. Since this is an alternative function to an I/O port, the corresponding data direction control bit must be set (one) to control an output pin. The control configuration is defined in the following table:

Table 4: Compare 0 Mode Select

COM01	COM00	Description
0	0	Timer/Counter0 disconnected from output pin OC0.
0	1	Toggle the OC0 output line.
1	0	Clear the OC0 output line (to zero).
1	1	Set the OC0 output line (to one).

Note: When changing the COM01/COM00 bits, Output Compare Interrupt 0 must be disabled by clearing its Interrupt Enable bit in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bit 3 - CTC0 : Clear Timer/Counter0 on Compare match:
When the CTC0 control bit is set (one), the Timer/Counter0 is reset to $00 in the clock cycle after the compare match. If the CTC0 control bit is cleared, the Timer/Counter0 continues running freely until it is stopped, set, cleared or wraps around (overflow).

Bits 2,1,0 - CS02, CS01, CS00 : Clock Select0, bit 2,1 and 0:
The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer0.

Table 5: Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T0, rising edge
1	1	1	External Pin T0, falling edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual data direction control register (cleared to zero gives an input pin).

Bits 5..3 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and always read zero.

THE TIMER COUNTER 0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
$32	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.

THE OUTPUT COMPARE REGISTER 0 - OCR0

Bit	7	6	5	4	3	2	1	0	
$31	MSB							LSB	OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Output Compare Register 0 is the source register for the Timer/Counter0 compare match functions.

The Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted from 16 to 2048 ms. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. Eight different clock cycle periods can be selected to determine the maximum period between two WDR instructions to avoid that the Watchdog Timer resets the MCU. If the reset period expires without another WDR instruction, the AT90S1300 resets and executes from the reset vector.

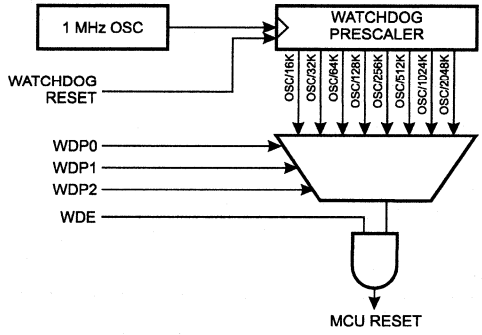


Figure 14: Watchdog Timer

THE WATCHDOG TIMER CONTROL REGISTER - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21	-	-	-	-	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..4 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and will always read as zero.

Bit 3 - WDE : Watchdog Enable:
When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled.

Bits 2..0 - WDP2..0 : Watchdog Timer Prescaler 2,1 and 0:
The WDP2..0 determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 6.

Table 6: Watchdog Timer Prescale Select

WDP2	WDP1	WDP0	Timeout Period
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space using the IN and OUT instructions.

The write access time is in the range of 2.5 - 4ms, depending on the Vcc voltages. A self-timing function, however, lets the user software detect when the next byte can be written.

The read access time is the same as for the Flash memory and is of no concern to the user software.

THE EEPROM ADDRESS REGISTER - EEAR

Bit	7	6	5	4	3	2	1	0	
$1E	-	MSB						LSB	EEAR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - Res : Reserved bit:
This bit is a reserved bit in the AT90S1300 and will always read as zero.

Bits 6..0 - EEAR6..0 : EEPROM Address:
The EEPROM Address Register - EEAR6..0 - specifies the EEPROM address in the 128 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 127.

THE EEPROM DATA REGISTER - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..0 - EEDR7..0 : EEPROM Data:
For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

THE EEPROM CONTROL REGISTER - EECR

Bit	7	6	5	4	3	2	1	0	
$1C	-	-	-	-	-	-	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..2 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and will always be read as zero.

Bit 1 - EEWE : EEPROM Write Enable:
The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. When the write access time (typically 2.5ms at Vcc=5V and 4ms at Vcc=2.7V) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte.

Bit 0 - EERE : EEPROM Read Enable:
The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access time is within a single clock cycle and there is no need to poll the EERE bit.

The Analog Comparator

The analog comparator compares the input values on the positive pin PB0 (AIN0) and the negative pin PB1 (AIN1). When the voltage on the positive pin PB0 (AIN0) is higher than the voltage on the negative pin PB1 (AIN1), the Analog Comparator Output, ACO is set (one). The comparator's output can be set to trigger the Analog Comparator interrupt. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 15.

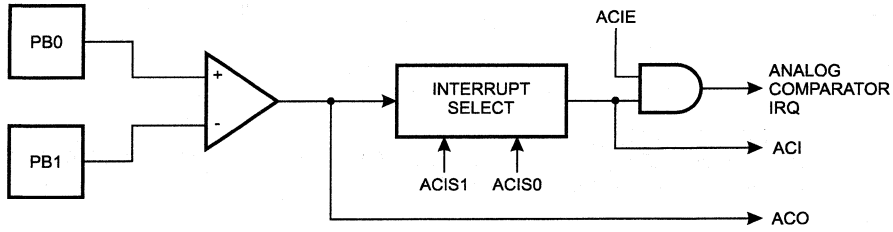


Figure 15: Analog Comparator Block Diagram

THE ANALOG COMPARATOR CONTROL AND STATUS REGISTER - ACSR

Bit	7	6	5	4	3	2	1	0	
$08	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..6 - Res : Reserved bits:
These bits are reserved bits in the AT90S1300 and will always read as zero.

Bit 5 - ACO : Analog Comparator Output:
ACO is directly connected to the comparator output.

Bit 4 - ACI : Analog Comparator Interrupt Flag:
This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by

hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

Bit 3 - ACIE : Analog Comparator Interrupt Enable:
When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled. For details on the comparator, refer to Page 4-22.

Bit 2 - ACIC : Analog Comparator Input Capture enable:
When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the analog comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

Bits 1,0 - ACIS1, ACIS0 : Analog Comparator Interrupt Mode Select:
These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 7.

Table 7: ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

I/O-Ports

Port B

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

Three data memory address locations are allocated for the Port B, one each for the Data Register - PORTB ($18), Data Direction Register - DDRB ($17) and the Port B Input Pins - PINB ($16). The Port B Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pullups. The Port B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

The Port B pins with alternate functions are shown in the following table:

Table 8: Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	AIN0 (Analog comparator positive input)
PB1	AIN1 (Analog comparator negative input)
PB2	OC0 (Timer/Counter0 Output compare match output)
PB5	MOSI (Data input line for memory downloading)
PB6	MISO (Data output line for memory uploading)
PB7	SCK (Master clock input)

When the pins are used for the alternate function the DDRB and PORTB register has to be set according to the alternate function description.

THE PORTB DATA REGISTER - PORTB

Bit	7	6	5	4	3	2	1	0	
$18	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B DATA DIRECTION REGISTER - DDRB

Bit	7	6	5	4	3	2	1	0	
$17	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B INPUT PIN ADDRESS - PINB

Bit	7	6	5	4	3	2	1	0	
$16	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the PORTB Data Latch is read, and when reading PINB, the logical values present on the pins are read.

PORTB AS GENERAL DIGITAL I/O

All 8 bits in port B are equal when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PBn is set (one) and the pin is configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PBn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 9: DDBn Effect on PORTB Pins

DDBn	PBn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PBn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTB

The alternate pin functions of Port B are:

SCK - PORTB, Bit 7:

SCK , Clock input pin for Memory downloading.

MISO - PORTB, Bit 6:

MISO, Data output pin for Memory downloading.

MOSI - PORTB, Bit 5:

MOSI , Data input pin for Memory downloading.

OC0 - PORTB, Bit 2:

OC0, Output compare match output: The PB2 pin can serve as an external output when Timer/Counter0 compare matches. The PB2 pin has to be configured as an output (DDB2 is set (one)) to serve this function. See the timer description for further details, and how to enable the output.

AIN1 - PORTB, Bit 1:

AIN1, Analog Comparator Positive Input. When configured as an input (DDB1 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB1 is cleared (zero)), this pin also serves as the negative input of the on-chip analog comparator.

AIN0 - PORTB, Bit 0:

AIN0, Analog Comparator Positive Input. When configured as an input (DDB0 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB0 is cleared (zero)), this pin also serves as the positive input of the on-chip analog comparator.

PORT B SCHEMATICS

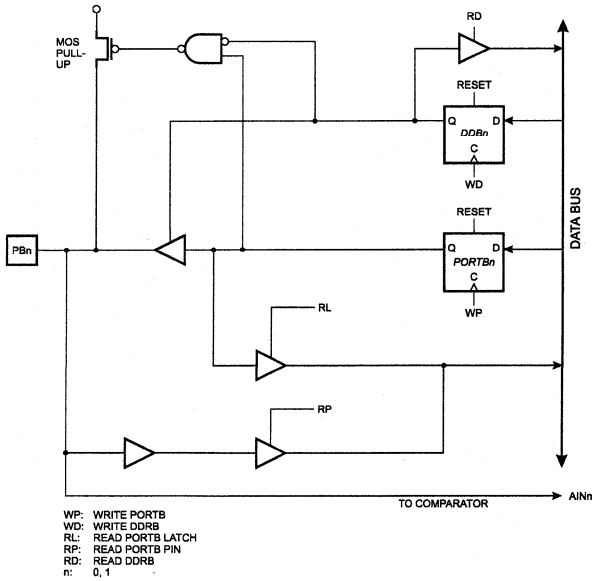


Figure 16: PORTB Schematic Diagram (pins PB0 and PB1)

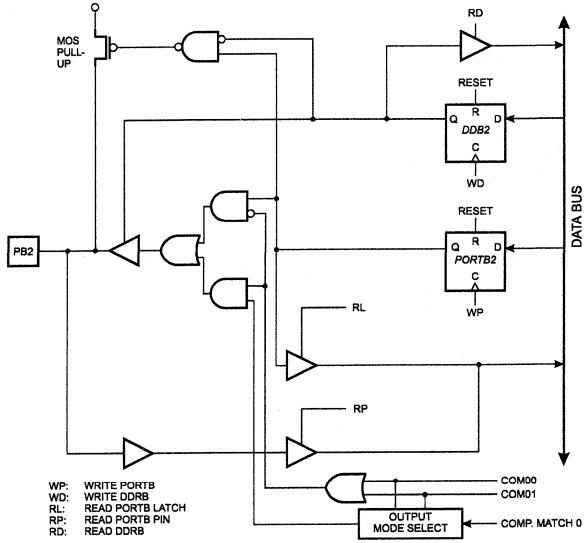


Figure 17: PORTB Schematic Diagram (Pin PB2)

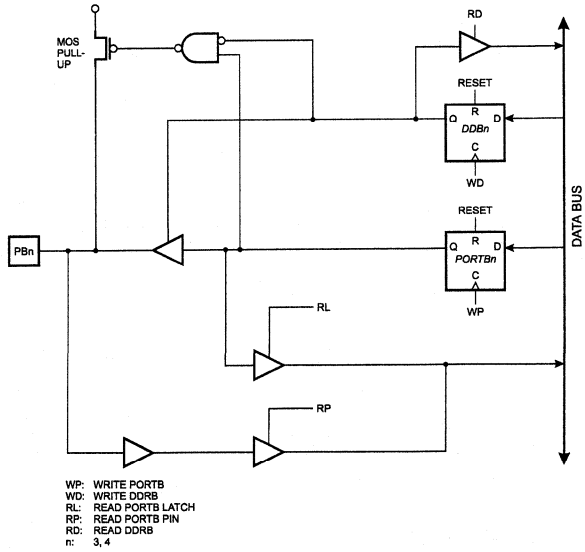


Figure 18: PORTB Schematic Diagram (Pins PB3 and PB4)

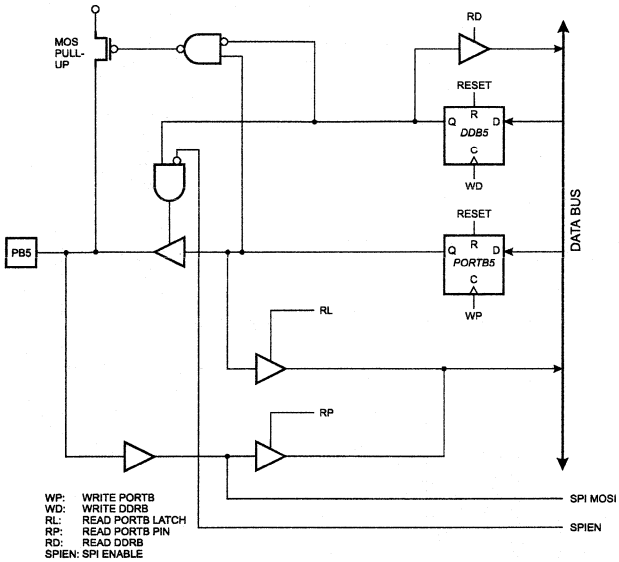


Figure 19: PORTB Schematic Diagram, Pin PB5

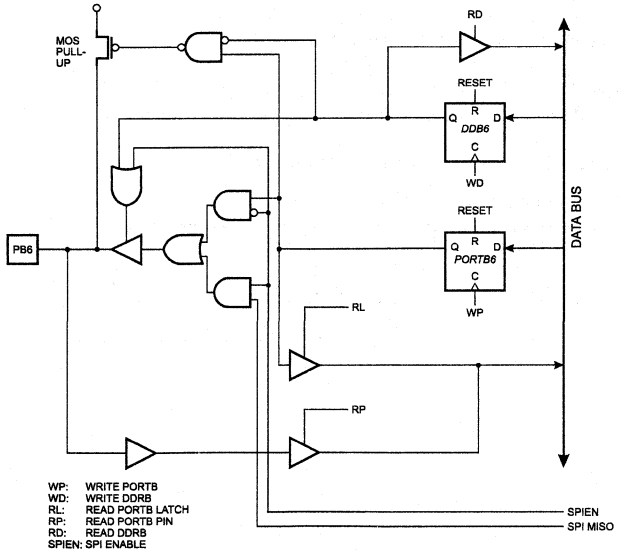


Figure 20: PORTB SchematicDiagram,in PB6

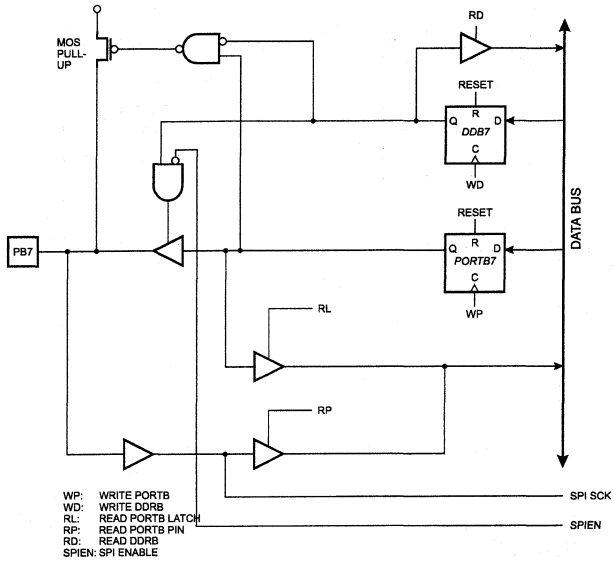


Figure 21: PORTB Schematic Diagram, Pin PB7

Port D

Three data memory address locations are allocated for the Port D, one each for the Data Register - PORTD ($12), Data Direction Register - DDRD ($11) and the Port D Input Pins - PIND ($10). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

Port D has seven bi-directional I/O pins with internal pullups, PD6..PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Some Port D pins have alternate functions as shown in the following table:

Table 10: Port D Pins Alternate Functions

Port Pin	Alternate Function
PD2	INT0 (External interrupt 0 input)
PD4	T0 (Timer/Counter 0 external input)

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alternate function description.

THE PORTD DATA REGISTER - PORTD

Bit	7	6	5	4	3	2	1	0	
$12	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D DATA DIRECTION REGISTER - DDRD

Bit	7	6	5	4	3	2	1	0	
$11	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D INPUT PINS ADDRESS - PIND

Bit	7	6	5	4	3	2	1	0	
$10	-	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the PORTD Data Latch is read, and when reading PIND, the logical values present on the pins are read.

PORTD AS GENERAL DIGITAL I/O

PDn, General I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when DDDn is configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PDn bit has to be cleared (zero) or the pin has to be configured as an output pin.

Table 11: DDDn Bits Effect on Port D Pins

DDDn	PDn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTD

The alternate functions of Port D are:

T0 - PORTD, Bit 4:

T0, Timer/Counter0 clock source: The PD4 pin has to be configured as an input (DDD4 is cleared (zero)) to serve this function. See the Timer description for further details. The internal pull up MOS resistor can be activated as described above.

INT0 - PORTD, Bit 2:

INT0, External Interrupt source 0: The PD2 pin can serve as an external active low interrupt source to the MCU. The PD2 pin has to be configured as an input (DDD0 is cleared (zero)) to serve this function. The internal pull up MOS resistor can be activated as described above. See the interrupt description for further details, and how to enable the source.

PORTD SCHEMATICS

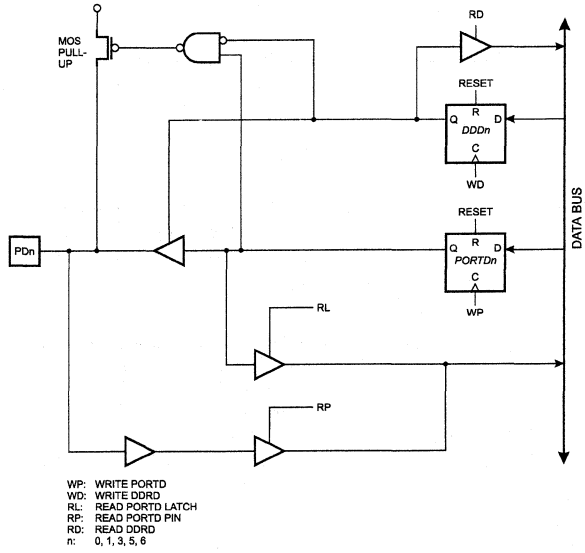


Figure 22: PORTD Schematic Diagram (Pins PD0, PD1, PD3, PD5 and PD6)

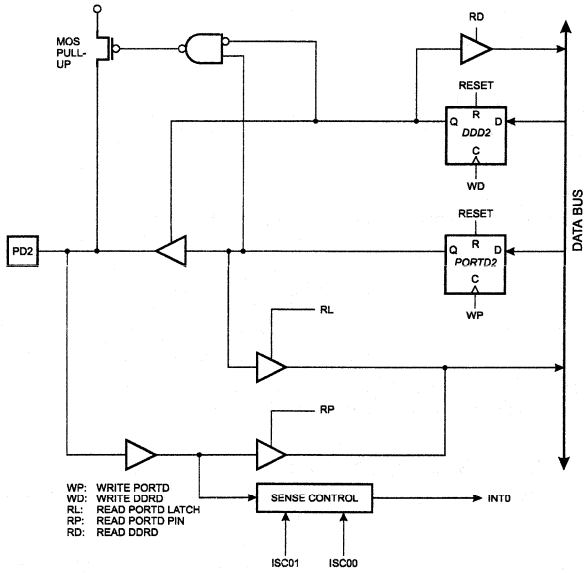


Figure 23: PORTD Schematic Diagram (Pin PD2)

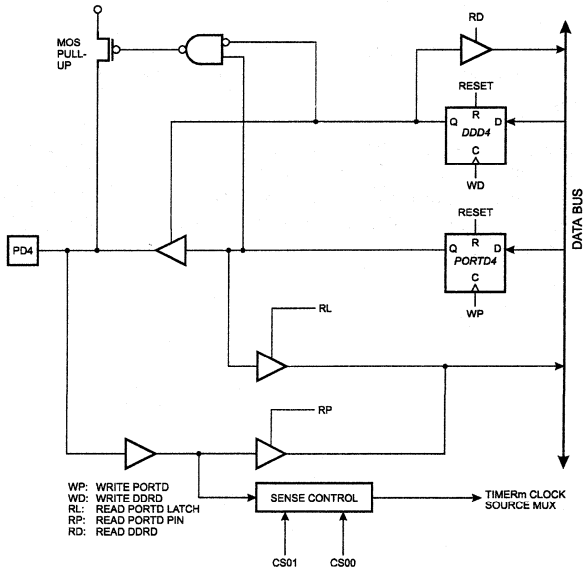


Figure 24: PORTD Schematic Diagram (Pin PD4)

Memory Programming

Program Memory Lock Bits

The AT90S1300 MCU provides two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 12.

Table 12: Lock Bit Protection Modes

Program Lock Bits			Protection Type
Mode	LB1	LB2	
1	U	U	No program lock features
2	P	U	Further programming of the Flash is disabled
3	P	P	Same as mode 2, but verify is also disabled.

Note: The Lock Bits can only be erased with the Chip Erase operation.

Programming the Flash and EEPROM

Atmel's AT90S1300 offers 1K bytes of in-system reprogrammable Flash PEROM Program memory and 128 bytes of EEPROM Data memory.

The AT90S1300 is normally shipped with the on-chip PEROM Program memory and EEPROM Data memory arrays in the erased state (i.e. contents = $FF) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The serial programming mode provides a convenient way to download the Program and Data into the AT90S1300 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Program and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one continuos address space: $000 to $3FF for the Program array and $400 to $47F for the Data array.

The Program and Data memory arrays on the AT90S1300 are programmed byte-by-byte in either programming modes. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode. In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming

INTERNAL ADDRESS COUNTER

The AT90S1300 contains an internal Flash address counter which is always reset to $000 on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

PARALLEL PROGRAMMING ALGORITHM

All major programming vendors offer worldwide support for the Atmel microcontroller series. To program and verify the AT90S1300 in the parallel programming mode, the following sequence is recommended:

1. *Power-up sequence:*
   Apply power between VCC and GND pins. Set RST and XTAL1 to GND.
   With all other pins floating, wait for more than 10 ms.
2. Set RST to 'H'.
   Set PD2 to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins PD3, PD4, PD5 to select one of the programming operations shown in Table 13
4. *Programming and verifying:*
   Apply data for Program memory location $000 to PB0 - PB7.
5. Raise RST to 12V to enable programming.
6. Pulse PD2 once to program a byte in the Program memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins PD3 to PD6 to the appropriate levels. Output data can be read at the port PB pins.
8. To program a byte at the next address location, pulse the XTAL1 pin once to advance the internal address counter. Apply new data to the port PB pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2KB array + 128 bytes of EEPROM or until the end of the object file is reached.
10. *Power-off sequence:*
    Set XTAL1 to 'L'.
    Set RST to 'L'.
    Float all other I/O pins.
    Turn VCC power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

$\overline{\text{DATA}}$ POLLING

The AT90S1300 features $\overline{\text{DATA}}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel programming mode, an attempted read of the last byte written will result in the complement of the written datum on PB7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{DATA}}$ Polling may begin any time after a write cycle has been initiated.

READY/$\overline{\text{BUSY}}$

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin PD1 is pulled Low after PD2 goes High during programming to indicate $\overline{\text{BUSY}}$. PD1 is pulled High again when programming is done to indicate READY.

PROGRAM VERIFY

Program Verify: If lock bits LB1 and LB2 have not been programmed, Flash Program Memory can be read back via the data lines for verification:

1. Reset the internal address counter to $000 by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Program data and read the output data at the port PB pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next Program data byte at the port PB pins.
5. Repeat steps 3 and 4 until the entire array is read. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

CHIP ERASE

The entire Flash array (1K bytes) + EEPROM (128 bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding PD2 low for 10 ms. The Program array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

READING THE SIGNATURE BYTES

The signature bytes are read by the same procedure as a normal verification of locations $000 and $001, except that PD5 and PD6 must be pulled to a logic low. The values returned are as follows.
($000) = $1E indicates manufactured by Atmel.
($001) = $90 indicates 1K bytes Program Flash Memory.
($002) = $01 indicates 90S1300 device when ($001) = $90.

Table 13: Flash and EEPROM Parallel Programming Modes

Mode	RST	PD2/ PROG (1)	PD3	PD4	PD5	PD6	Data I/O PB7:0
Chip Erase	12V	⎺_/⎺	H	L	L	L	X
Write Memory(2,3)	12V	⎺_/⎺	L	H	H	H	DIN
Read Memory(2)	H	H	L	L	H	H	DOUT
Write Lock Bit 1	12V	⎺_/⎺	H	H	H	H	X
Write Lock Bit 2	12V	⎺_/⎺	H	H	L	L	X
Read Signature Byte	H	H	L	L	L	L	DOUT
Serial Prog. Enable	H	(1) ⎺_/⎺	L	H	L	H	PB0=0
Serial Prog. Disable	H	(1) ⎺_/⎺	L	H	L	H	PB0=1
Read Serial Prog. Fuse	H	L	H	H	L	H	PB0=SPF

Notes:

1. Chip erase requires a 10 ms $\overline{\text{PROG}}$ pulse.
2. The internal Flash address counter is reset to $000 on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.
3. PD1 is pulled Low during programming to indicate RDY/$\overline{\text{BSY}}$.

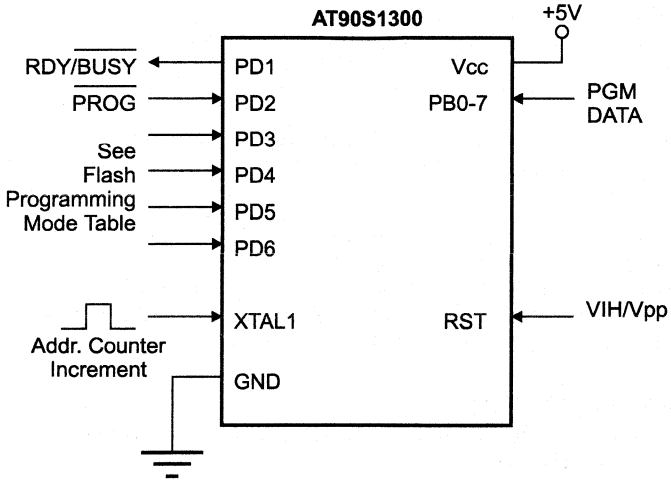


Figure 25: Parallel Programming

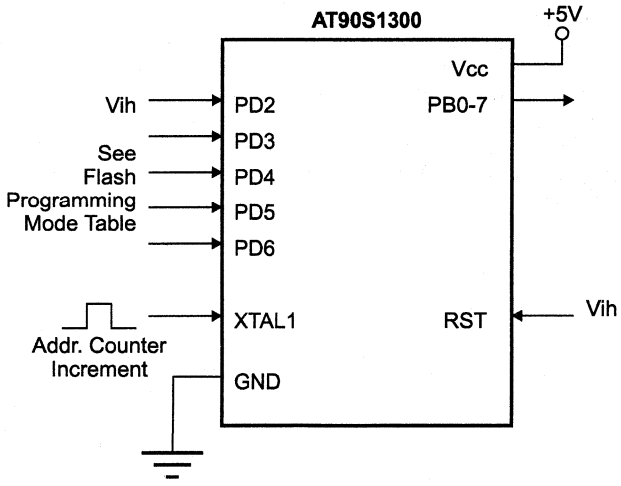


Figure 26: Parallel Verify

Serial Downloading

Both the Program and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to VCC. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and Data arrays into $FF.

The Program and Data memory arrays have separate address spaces:
$000 to $3FF for Program memory and $000 to $07F for Data memory.
Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 10 MHz oscillator clock, the maximum SCK frequency is 250 kHz.

SERIAL PROGRAMMING ALGORITHM

To program and verify the AT90S1300 in the serial programming mode, the following sequence is recommended (See three byte instruction formats in Table 14):

1. *Power-up sequence:*
   Apply power between VCC and GND.
   Set RST pin to 'H'.
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 1 MHz to 10 MHz clock to the XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB5. The frequency of the shift clock supplied at pin SCK/PB7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Program or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB6.
   $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
5. At the end of the programming session, RST can be set low to commence normal operation.
6. *Power-off sequence (if needed):*
   Set XTAL1 to 'L' (if a crystal is not used).
   Set RST to 'L'
   Float all other I/O pins.
   Turn Vcc power off.

Table 14: Serial Programming Instruction Set

Instruction	Instruction Format			Operation
	Byte 1	Byte 2	Byte3	
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable Serial Programming after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 1K & 128 byte memory arrays
Read Program Memory	000a a001	bbbb bbbb	oooo oooo	Read data **o** from Program memory at address **a:b**
Write Program Memory	000a a010	bbbb bbbb	iiii iiii	Write data **i** to Program memory at address **a:b**
Read Data Memory	0000 0101	xbbb bbbb	oooo oooo	Read data **o** from Data memory at address **b**
Write Data Memory	0000 0110	xbbb bbbb	iiii iiii	Write data **i** to Data memory at address **b**
Write Lock Bits	1010 1100	0*12*x x111	xxxx xxxx	Write lock bits. Set bits ***1,2***='0' to program lock bits.

Note: **a** = address high bits
b = address low bits
o = data out
i = data in
x = don't care
1 = lock bit 1
2 = lock bit 2

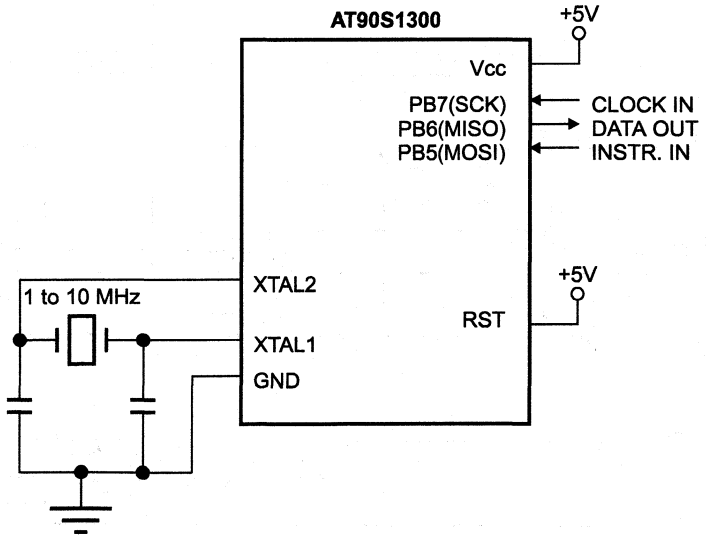


Figure 27: Serial Programming and Verify

When *writing* serial data to the AT90S1300, data is clocked on the *rising* edge of CLK.
When *reading* data from the AT90S1300, data is clocked on the *falling* edge of CLK. See Figure 29 for an explanation.

Programming Characteristics

Table 15: Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, Vcc = 5.0V±10%.

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	PD4 ($\overline{ENABLE}$)High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$ Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

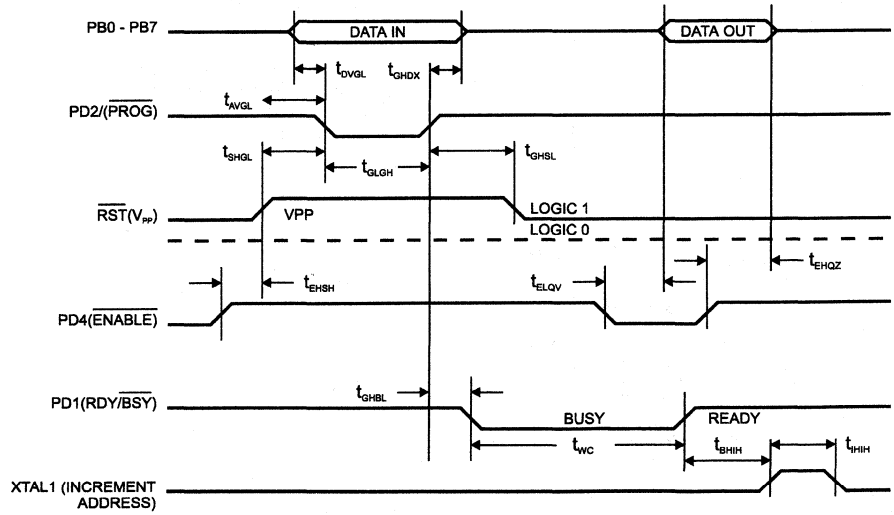


Figure 28: Flash/EEPROM Programming and Verfication Waveforms - Parallel Mode

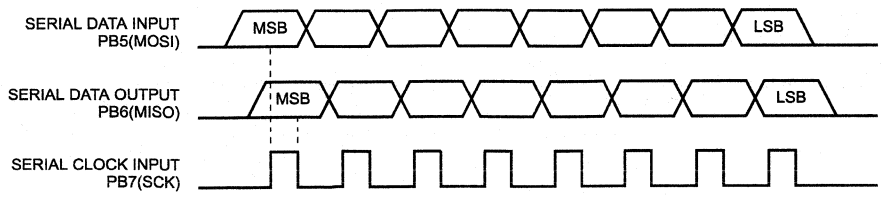


Figure 29: Serial Downloading Waveforms

Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin with respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	25.0 mA

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

D.C. Characteristics

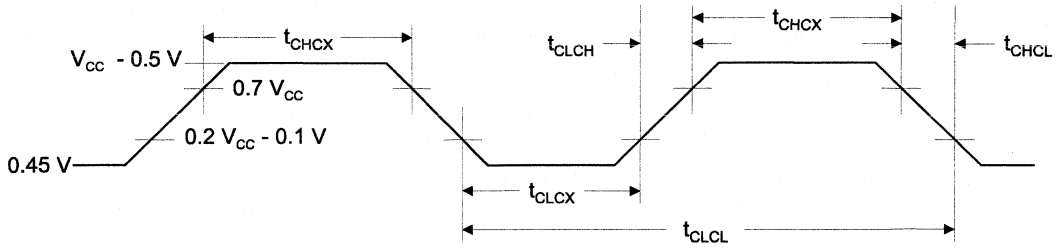
T_A=-40°C to 85°C Vcc=2.7V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		-0.5		0.2 Vcc - 0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RESET)	0.2 VCC + 0.9		VCC + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RESET)	0,7 VCC		VCC + 0.5	V
V_{OL}	Output Low Voltage[1] (Ports B, D)	I_{OL} = 20 mA, VCC = 5 V I_{OL} = 10 mA, VCC = 2.7 V			0.5	V
V_{OH}	Output High Voltage (Ports B,D)	I_{HI} = 10 mA, VCC = 5 V I_{HI} = 5 mA, VCC = 2.7 V	4.5			V
I_{OH}	Output Source Current (Ports B,D)	VCC = 5 V VCC = 2.7 V			10 5	mA
I_{OL}	Output Sink Current (Port B,D)	VCC = 5 V VCC = 2.7 V			20 10	mA
RRST	Reset Pulldown Resistor		10		50	KΩ
I_{CC}	Power Supply Current	Active Mode, 3V, 4MHz Idle Mode 3V, 4MHz		1.5 400		mA uA
I_{CC}	Power Down Mode[2]	WDT enabled, 3V WDT disabled, 3V		50 <1		µA µA

Notes:

1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 20mA
   Maximum total I_{OL} for all output pins: 80mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	VCC = 2.7 V to 6.0 V		VCC = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	10	0	24	MHz
t_{CLCL}	Clock Period	100		41.7		ns
t_{CHCX}	High Time	40		16.7		ns
t_{CLCX}	Low Time	40		16.7		ns
t_{CLCH}	Rise Time		10		4.15	ns
t_{CHCL}	Fall Time		10		4.15	ns

Ordering Information

Ordering Code	Package	Operation Range
AT90S1300-PC	20P3	Commercial
AT90S1300-SC	20S	(0°C to 70°C)
AT90S1300-PI	20P3	Industrial
AT90S1300-SI	20S	(-40°C to 85°C)

Package Type	
20P3	20 Lead, 0.300" Wide Plastic Dual In-Line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

AT90S1300 Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F	SREG	I	T	O	S	V	N	Z	C	2-12
$3E	Reserved									
$3D	Reserved									
$3C	Reserved									
$3B	GIMSK	-	INT0	-	-	-	-	-	-	2-14
$3A	Reserved									
$39	TIMSK	-	-	-	-	-	-	TOIE0	OCIE0	2-15
$38	TIFR							TOV0	OCF0	2-15
$37	Reserved									
$36	Reserved									
$35	MCUCR	-	-	SE	SM	-	-	ISC01	ISC00	2-16
$34	Reserved									
$33	TCCR0	TOV0	OCF0	COM01	COM00	CTC0	CS02	CS01	CS00	2-18
$32	TCNT0	Timer/Counter0 (8 Bit)								2-19
$31	OCR0	Timer/Counter0 Output Compare Register								2-20
$30	Reserved									
$2F	Reserved									
$2E	Reserved									
$2D	Reserved									
$2C	Reserved									
$2B	Reserved									
$2A	Reserved									
$29	Reserved									
$28	Reserved									
$27	Reserved									
$26	Reserved									
$25	Reserved									
$24	Reserved									
$23	Reserved									
$22	Reserved									
$21	WDTCR	-	-	-	-	WDE	WDP2	WDP1	WDP0	2-20
$20	Reserved									
$1F	Reserved									
$1E	EEAR	-	EEPROM Address Register							2-21
$1D	EEDR	EEPROM Data Register								2-21
$1C	EECR	-	-	-	-	-	-	EEWE	EERE	2-21
$1B	Reserved									
$1A	Reserved									
$19	Reserved									
$18	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	2-24
$17	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	2-24
$16	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	2-24
$15	Reserved									
$14	Reserved									
$13	Reserved									
$12	PORTD	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0	2-29
$11	DDRD	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	2-29
$10	PIND	-	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	2-29
$0F	Reserved									
$0E	Reserved									
$0D	Reserved									
$0C	Reserved									
$0B	Reserved									
$0A	Reserved									
$09	Reserved									
$08	ACSR	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	2-22
...	Reserved									
$00	Reserved									

AT90S1300 Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • (FFh - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC + k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC + k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

(continued)

AT90S1300 Instruction Set Summary (Continued)

Mnem-	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
BIT AND BIT-TEST INSTRUCTIONS					
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watch Dog Reset	(see specific descr. for WDR/timer)	None	1

Contents

Features

- Utilizes the AVR™ Enhanced RISC Architecture
- AVR™ - High Performance and Low Power RISC Architecture
- 112 Powerful Instructions - Most Single Clock Cycle Execution
- Efficient Context Switching Concept
- 2 Kbytes of In-System Reprogrammable Downloadable Flash
  SPI Serial Interface for Program Downloading
  Endurance: 1,000 Write/Erase Cycles
- 128 bytes EEPROM
  Endurance: 100,000 Write/Erase Cycles
- 64 bytes Internal RAM
- 32 x 8 General Purpose Working Registers
- 15 Programmable I/O Lines
- VCC Min.: 2.7 V
- Fully Static Operation
- One 8-Bit Timer/Counter with Separate Prescaler
- One 16-Bit Timer/Counter with Separate Prescaler and Compare and Capture Modes
- Full Duplex UART
- 10 bit PWM
- External and Internal Interrupt Sources
- Programmable Watchdog Timer
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Programming lock for Software Security
- 20-Pins Device

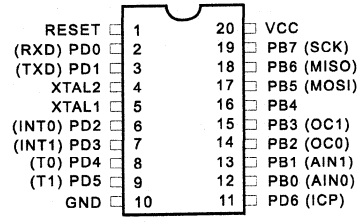
8-Bit AVR Microcontroller with 2K bytes Downloadable Flash

Description

The AT90S2312 is a low-power CMOS 8 bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S2312 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

(continued)

Pin Configuration



Block Diagram

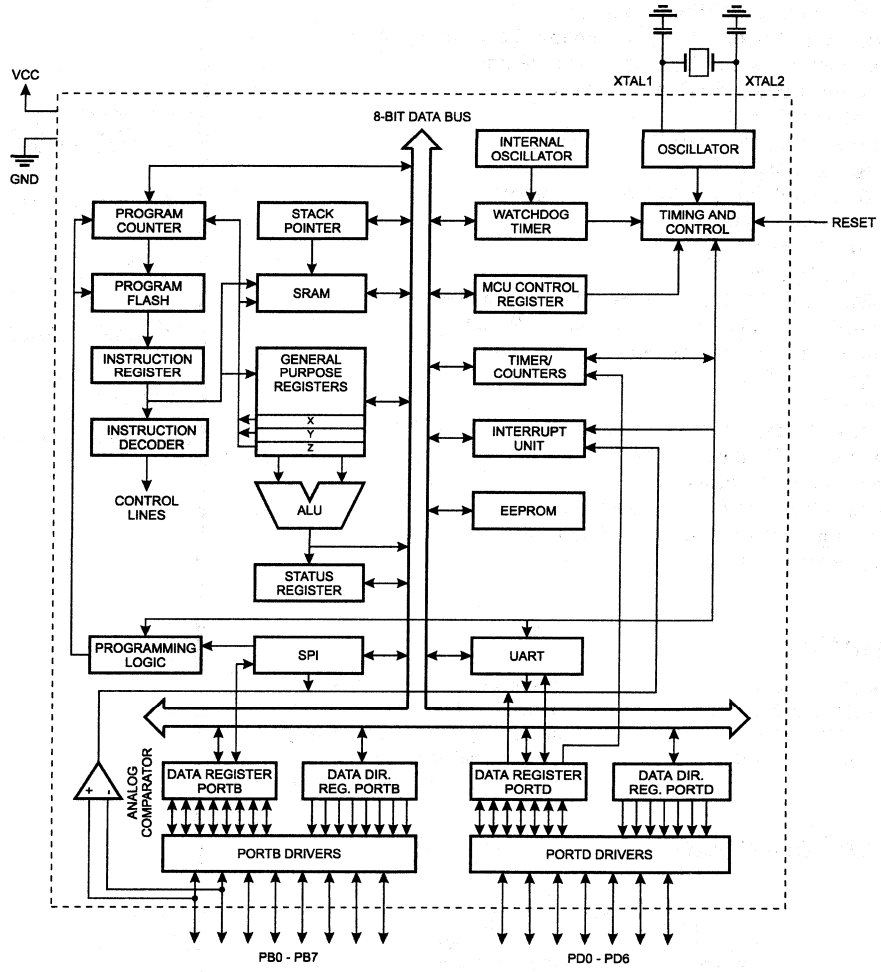


Figure 1: The AT90S2312 Block Diagram

Description (Continued)

The *AVR* core is based on an enhanced RISC architecture that combines a rich instruction set with the 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (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 architecture supports high level languages efficiently while on-chip context switching hardware allows high performance, low power real timer control system design.

The AT90S2312 provides the following features: 2K bytes of Downloadable Flash, 128 bytes EEPROM, 64-bytes SRAM, 15 general purpose I/O lines, 32 general purpose working registers, flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, programmable Watchdog Timer with internal oscillator, an SPI serial port for Flash Memory downloading and two software selectable power saving modes. The Idle Mode stops the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The power down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8 bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT90S2312 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90S2312 *AVR* is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC
Supply voltage pin.

GND
Ground pin.

Port B (PB7..PB0)
Port B is an 8-bit bi-directional I/O port. Port pins can provide internal pullups (selected for each bit). PB0 and PB1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip analog comparator. The Port B output buffers can sink 20mA and can drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

Port B also serves the functions of various special features of the AT90S2312 as listed on Page 3-45.

Port D (PD6..PD0)
Port D has seven bi-directional I/O pins with internal pullups, PD6..PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Port D also serves the functions of various special features of the AT90S2312 as listed on Page 3-51.

RESET
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier

Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively , of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

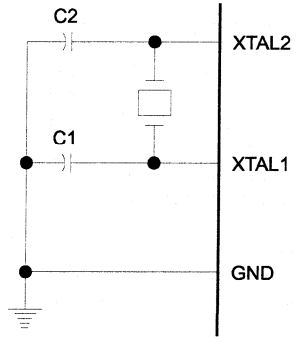


Figure 2: Oscillator Connections

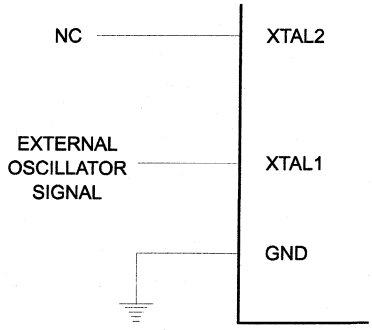


Figure 3: External Clock Drive Configuration

AT90S2312 *AVR* Enhanced RISC Microcontroller CPU

The AT90S2312 *AVR* RISC microcontroller is upward compatible with the *AVR* Enhanced RISC Architecture. The programs written for the AT90S2312 MCU are fully compatible with the range of *AVR* 8-bit MCUs (AT90Sxxxx) with respect to source code and clock cycles for execution.

Architectural Overview

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 ALU (Arithmetic Logic Unit) 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-bits indirect address register pointers for SRAM addressing - enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look up function. These added function registers are the 16-bits X-register, Y-register and Z-register.

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S2312 *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 lowermost SRAM addresses, allowing them to be accessed as though they were ordinary memory locations.

The *AVR* has Harvard architecture - with separate memories and buses for program and data. The program memory is accessed 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 in-system downloadable Flash memory.

With the relative jump and call instructions, the whole 1K address space is directly accessed. All *AVR* instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-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 SP in the reset routine (before subroutines or interrupts are executed). The 16-bit stack pointer SP is read/write accessible in the I/O space.

The 64 bytes data SRAM can be easily accessed through the four different addressing modes supported in the *AVR* architecture.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, A/D-converters, and other I/O functions. The memory spaces in the *AVR* architecture are all linear and regular memory maps.

AVR™ AT90S2312 Architecture

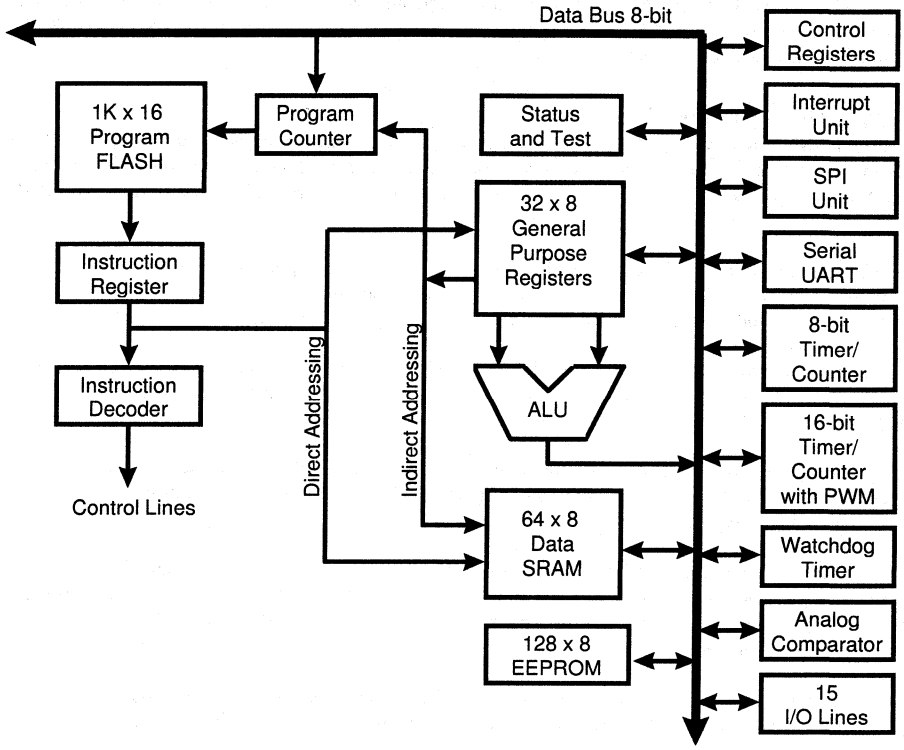


Figure 4: The AT90S2312 *AVR* Enhanced RISC Architecture

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 the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt address vector the higher priority.

The General Purpose Register File

Figure 5 shows the structure of the 32 general purpose registers in the CPU.

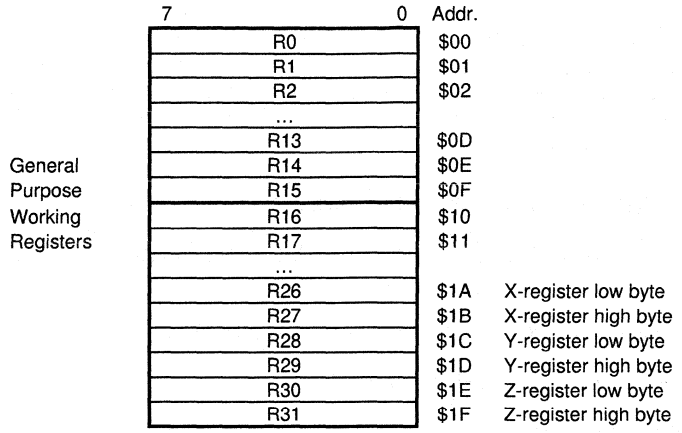


Figure 5: *AVR* CPU general purpose working registers

All the 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, 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, OR and all other operations between two registers or on a single register apply to the entire register file.

As shown Figure 5, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user SRAM area. 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.

The 64 bytes of SRAM available for general data are implemented as addresses $20 to $5F.

THE X-REGISTER, Y-REGISTER AND Z-REGISTER

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

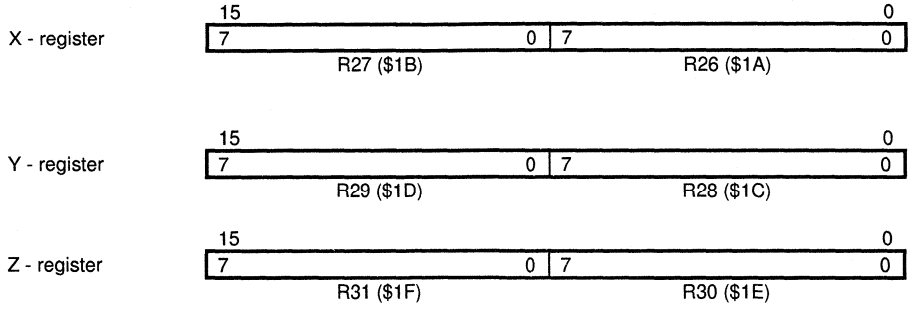


Figure 6: The X, Y and Z Registers

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

The ALU - Arithmetic Logic Unit

The high-performance *AVR* ALU operates in direct connection with all the 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, logic and bit-functions. Some microcontrollers in the *AVR* product family feature a hardware multiplier in the arithmetic part of the ALU.

The Downloadable Flash Program Memory

The AT90S2312 contains 2K bytes on-chip downloadable Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 1K x 16 words. The Flash memory has an endurance of at least 1000 write/erase cycles.

The AT90S2312 Program Counter PC is 10 bits wide, thus addressing the 1024 program memory addresses.

See Page 3-54 for a detailed description on Flash data downloading.

Constant tables must be allocated within the address 0-2K (see the LPM - Load Program Memory instruction description).

See Page 3-11 for the different addressing modes.

The EEPROM Data Memory

The AT90S2312 contains 128 bytes of EEPROM data memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on Page 3-34 specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register.

For the SPI data downloading, see Page 3-58 for a detailed description.

The SRAM Data Memory

The following figure shows how the AT90S2312 SRAM Memory is organized:

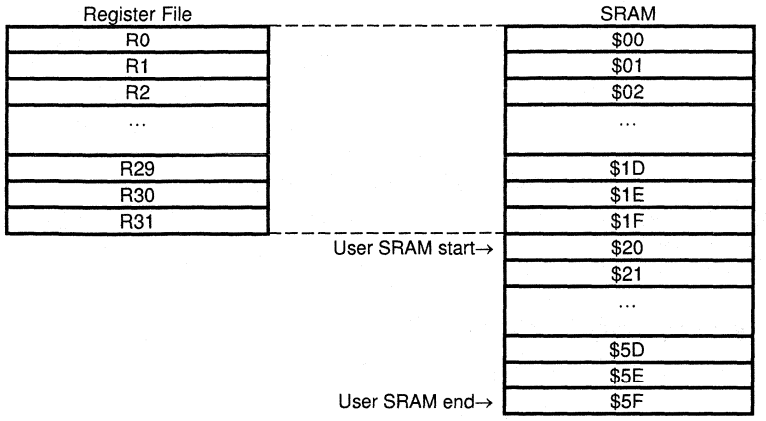


Figure 7: SRAM Organization

The 96 SRAM Memory locations address both the Register file and the data SRAM. The first 32 locations address the register file, and the next 64 locations address the data SRAM.

The four different addressing modes for the SRAM data memory cover: Direct 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.

The Direct with Displacement mode features 63 address locations reach from the base address given by the Y and Z register.

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

The 32 registers plus the 64 bytes of data SRAM in the AT90S2312 are all directly accessible through all these addressing modes.

See Page 3-11 for a detailed description of the different addressing modes.

The Program and Data Addressing Modes

The AT90S2312 *AVR* Enhanced RISC Microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM). This section describes the different addressing modes supported by the *AVR* architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

REGISTER DIRECT, SINGLE REGISTER Rd

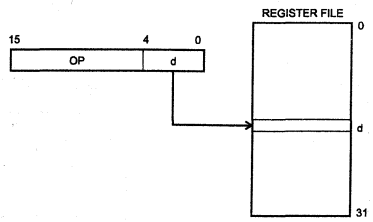


Figure 8: Direct Single Register Addressing

The operand is contained in register d (Rd).

REGISTER DIRECT, TWO REGISTERS Rd AND Rr

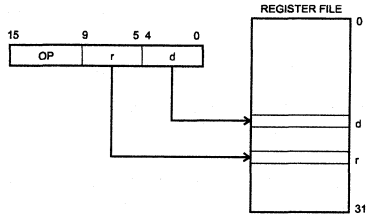


Figure 9: Direct Register Addressing, Two Registers

Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O DIRECT

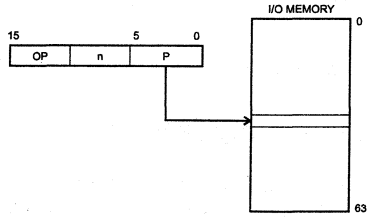


Figure 10: I/O Direct Addressing

Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

SRAM DIRECT WITH DISPLACEMENT

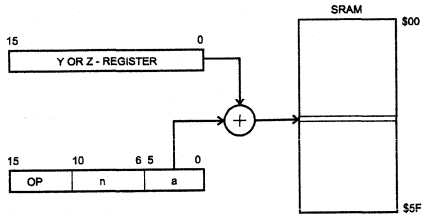


Figure 11: SRAM Direct with Displacement

Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.

SRAM/REGISTER INDIRECT

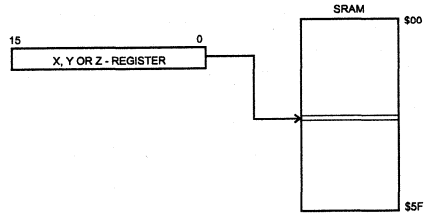


Figure 12: SRAM Indirect Addressing

Operand address is the contents of the X, Y or the Z-register.

SRAM/REGISTER INDIRECT WITH PRE-DECREMENT

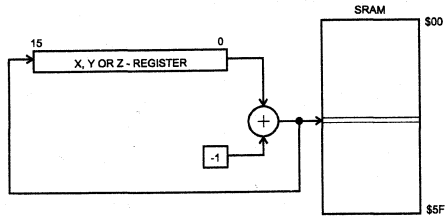


Figure 13: SRAM Indirect Addressing With Pre-Decrement

The X , Y or the Z-register is decremented before the operation. Operand address is the decremented contents of the X, Y or the Z-register.

SRAM/REGISTER INDIRECT WITH POST-INCREMENT

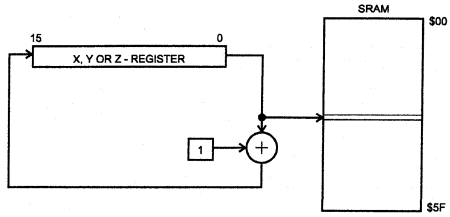


Figure 14: SRAM Indirect Addressing With Post-Increment

The X, Y or the Z-register is incremented after the operation. Operand address is the content of the X, Y or the Z-register prior to incrementing.

CONSTANT ADDRESSING USING THE LPM INSTRUCTION

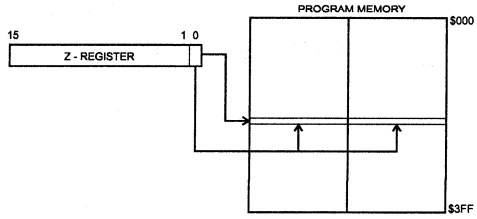


Figure 15: Code Memory Constant Addressing

Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 1K) and LSB, select low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

INDIRECT PROGRAM ADDRESSING, IJMP AND ICALL

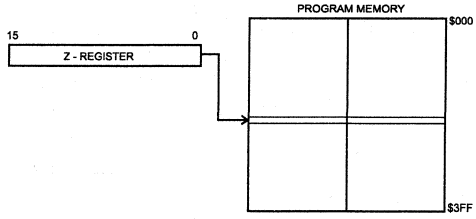


Figure 16: Indirect Program Memory Addressing

Program execution continues at address contained by the Z-register (i.e. the PC is loaded with the content of the Z-register).

RELATIVE PROGRAM ADDRESSING, RJMP AND RCALL

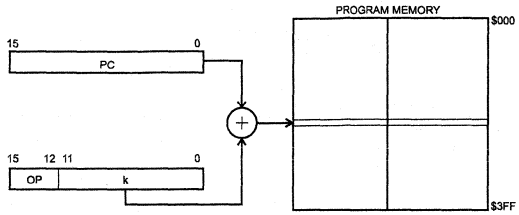


Figure 17: Relative Program Memory Addressing

Program execution continues at address PC + k. The relative address k is in the range from -2K to +(2K - 1).

Memory Access and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The *AVR* CPU is driven by the System Clock Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 18 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

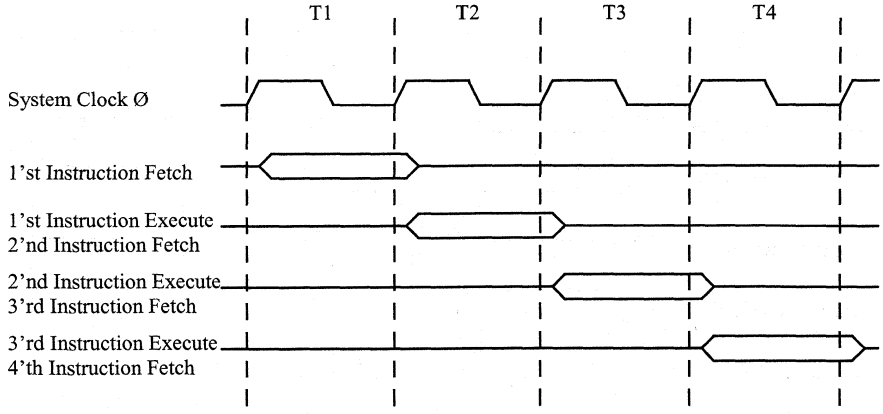


Figure 18: The Parallel Instruction Fetches and Instruction Executions

Figure 19 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

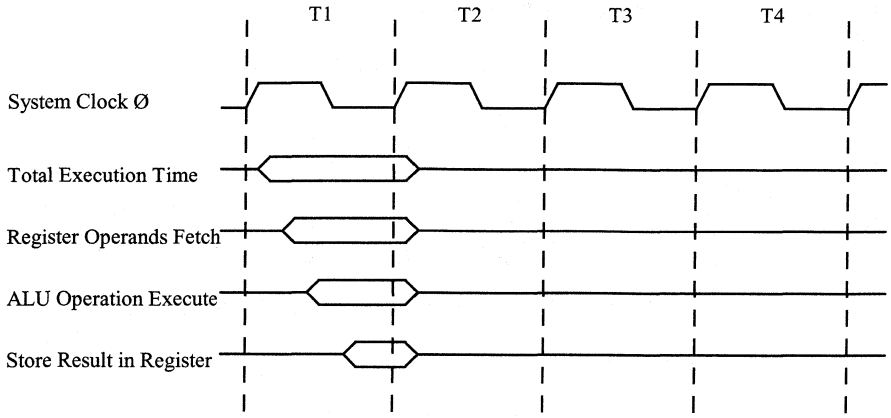


Figure 19: Single Cycle ALU Operation

The internal data SRAM access is performed in two System Clock cycles as described in Figure 20.

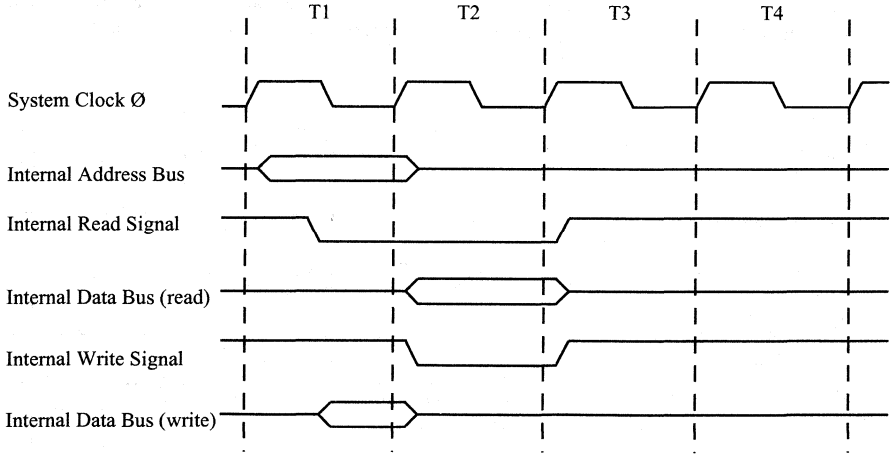


Figure 20: On-Chip Data SRAM Access Cycles

I/O Memory

The I/O space definition of the AT90S2312 is shown in the following table:

Table 1: AT90S2312 I/O Space

Address Hex	Name	Function
$3F	SREG	Status REGister
$3E	SPH	Stack Pointer High
$3D	SPL	Stack Pointer Low
$3B	GIMSK	General Interrupt MaSK register
$39	TIMSK	Timer/Counter Interrupt MaSK register
$38	TIFR	Timer/Counter Interrupt Flag register
$35	MCUCR	MCU general Control Register
$33	TCCR0	Timer/Counter 0 Control Register
$32	TCNT0	Timer/Counter 0 (8-bit)
$31	OCR0	Output Compare Register 0
$2F	TCCR1A	Timer/Counter 1 Control Register A
$2E	TCCR1B	Timer/Counter 1 Control Register B
$2D	TCNT1H	Timer/Counter 1 High Byte
$2C	TCNT1L	Timer/Counter 1 Low Byte
$2B	OCR1H	Output Compare Register 1 High Byte
$2A	OCR1L	Output Compare Register 1 Low Byte
$25	ICR1H	T/C 1 Input Capture Register High Byte
$24	ICR1L	T/C 1 Input Capture Register Low Byte
$21	WDTCR	Watchdog Timer Control Register
$1E	EEAR	EEPROM Address Register
$1D	EEDR	EEPROM Data Register
$1C	EECR	EEPROM Control Register
$18	PORTB	Data Register, Port B
$17	DDRB	Data Direction Register, Port B
$16	PINB	Input Pins, Port B
$12	PORTD	Data Register, Port D
$11	DDRD	Data Direction Register, Port D
$10	PIND	Input Pins, Port D
$0C	UDR	UART I/O Data Register
$0B	USR	UART Status Register
$0A	UCR	UART Control Register
$09	UBRR	UART Baud Rate Register
$08	ACSR	Analog Comparator Control and Status Register

Note: reserved and unused locations are not shown in the table

All the different AT90S2312 I/O and peripherals are placed in the I/O space. The different 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.

The different I/O and peripherals control registers are explained in the following sections.

THE STATUS REGISTER - SREG

The *AVR* status register - SREG - at I/O space location $3F is defined as:

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

Bit 7 - I : Global Interrupt Enable:
The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in the interrupt mask registers - GIMSK and TIMSK. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the GIMSK and TIMSK values. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

Bit 6 - T : Bit Copy Storage:
The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

Bit 5 - H : Half Carry Flag:
The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

Bit 4 - S : Sign Bit, $S = N \oplus V$:
The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

Bit 3 - V : Two's Complement Overflow Flag:
The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

Bit 2 - N : Negative Flag:
The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 1 - Z : Zero Flag:
The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 0 - C : Carry Flag:
The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information.

THE STACK POINTER - SP

The general *AVR* 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E and $3D. Since the stack address space is within the data SRAM, a 7 bit stack pointer is used to address the stack in the AT90S2312.

Bit	15	14	13	12	11	10	9	8	
$3E	-	-	-	-	-	-	-	-	SPH
$3D	-	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when data is pushed onto the Stack with subroutine CALL and interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt IRET.

Reset and Interrupt Handling

The AT90S2312 provides 11 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. All the 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 2. 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 External Interrupt Request 0 etc.

Table 2: Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	Hardware Pin and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT1	Timer/Counter1 Capture Event
5	$004	TIMER1 COMP1	Timer/Counter1 Compare Match
6	$005	TIMER1 OVF1	Timer/Counter1 Overflow
7	$006	TIMER0, COMP0	Timer/Counter0 Compare Match
8	$007	TIMER0, OVF0	Timer/Counter0 Overflow
9	$008	UART, RX	UART, Rx Complete
10	$009	UART, UDRE	UART Data Register Empty
11	$00A	UART, TX	UART, Tx Complete
12	$00B	ANA_COMP	Analog Comparator

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

```
Address     Labels      Code              Comments

$000                    rjmp  RESET       ; Reset Handle
$001                    rjmp  EXT_INT0    ; IRQ0 Handle
$002                    rjmp  EXT_INT1    ; IRQ1 Handle
$003                    rjmp  TIM_CAPT1   ; Timer1 capture Handle
$004                    rjmp  TIM_COMP1   ; Timer1 compare Handle
$005                    rjmp  TIM_OVF1    ; Timer1 overflow Handle
$006                    rjmp  TIM_COMP0   ; Timer0 compare Handle
$007                    rjmp  TIM_OVF0    ; Timer0 overflow Handle
$008                    rjmp  UART_RXC    ; UART RX Complete Handle
$009                    rjmp  UART_DRE    ; UDR Empty Handle
$00a                    rjmp  UART_TXC    ; UART TX Complete Handle
$00b                    rjmp  ANA_COMP    ; Analog Comparator Handle
;
$00c        MAIN:       <instr>  xxx      ; Main program start
 ...         ...          ...      ...
```

RESET

A Reset State is enabled by a high level on the RESET pin. The pin must be held high for at least two crystal clock cycles. All internal 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 RJMP - relative jump - 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.

INTERRUPT HANDLING

The AT90S2312 has two 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 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 register0 matching the value of Timer/Counter0) 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, hardware 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.

THE GENERAL INTERRUPT MASK REGISTER - GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B	INT1	INT0	-	-	-	-	-	-	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - INT1 : External Interrupt Request 1 Enable:

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control1 bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. If the INT1

pin for external interrupts shall be activated, the DDD3 bit in the Data Direction Register PORTD (DDRD) must be cleared (zero) to force an input pin. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address $002. See also "External Interrupts".

Bit 6 - INT0 : External Interrupt Request 0 Enable:
When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. If the INT0 pin for external interrupts shall be activated, the DDD2 bit in the Data Direction Register PORTD (DDRD) must be cleared (zero) to force an input pin. The corresponding interrupt of External Interrupt Request 0 is executed from program memory address $001. See also "External Interrupts".

Bits 5..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read as zero.

THE TIMER/COUNTER INTERRUPT MASK REGISTER - TIMSK

Bit	7	6	5	4	3	2	1	0	
$39	TOIE1	OCIE1	-	-	TICIE1	-	TOIE0	OCIE0	TIMSK
Read/Write	R/W	R/W	R	R	R/W	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - TOIE1 : Timer/Counter1 Overflow Interrupt Enable:
When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $005) is executed if an overflow in Timer/Counter1 occurs. The Overflow Flag (Timer/Counter1) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR. When Timer/Counter1 is in PWM mode, the Timer Overflow flag is set when the counter changes counting direction at $0000.

Bit 6 - OCIE1 :Timer/Counter1 Output Compare Match Interrupt Enable:
When the OCIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare Match interrupt is enabled. The corresponding interrupt (at vector $004) is executed if a Compare match in Timer/Counter1 occurs. The Compare Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 5,4 - Res :Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

Bit 3 - TICIE1 : Timer/Counter1 Input Capture Interrupt Enable:
When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector $003) is executed if a capture-triggering event occurs on pin 11, PD6(ICP). The Input Capture Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 2 - Res : Reserved bit:
This bit is a reserved bit in the AT90S2312 and always reads as zero.

Bit 1 - TOIE0 : Timer/Counter0 Overflow Interrupt Enable:
When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $007) is executed if an overflow in Timer/Counter0 occurs. The Overflow Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 0 - OCIE0 :Timer/Counter0 Output Compare Match Interrupt Enable:
When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at vector $006) is executed if a compare match in Timer/Counter0 occurs. The Compare Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

THE TIMER/COUNTER INTERRUPT FLAG REGISTER - TIFR

Bit	7	6	5	4	3	2	1	0	
$38	TOV1	OCF1	-	-	ICF1	-	TOV0	OCF0	TIFR
Read/Write	R/W	R/W	R	R	R/W	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - TOV1 : Timer/Counter1 Overflow Flag:
The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic one to the flag. When the I-bit in SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflow Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 changes counting direction at $0000.

Bit 6 - OCF1 : Output Compare Flag 1:
The OCF1 bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1 - Output Compare Register 1. OCF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1 is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1 (Timer/Counter1 Compare match Interrupt Enable), and the OCF1 is set (one), the Timer/Counter1 Compare match Interrupt is executed.

Bit s 5, 4 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

Bit 3 - ICF1 : - Input Capture Flag 1:
The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to the input capture register - ICR1. ICF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logic one to the flag.

Bit 2 - Res : Reserved bit:
This bit is a reserved bit in the AT90S2312 and always reads zero.

Bit 1 - TOV0 : Timer/Counter0 Overflow Flag:
The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

Bit 0 - OCF0 : Output Compare Flag 0:
The OCF0 bit is set (one) when a compare match occurs between the Timer/Counter0 and the compared data in OCR0 - Output Compare Register0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0 is cleared by writing a logic one to the flag.. When the SREG I-bit, and OCIE0 (Timer/Counter0 Compare match Interrupt Enable), and OCF0 are set (one), the Timer/Counter0 Compare match Interrupt is executed.

EXTERNAL INTERRUPTS

The external interrupts are triggered by the INT1 and INT0 pins. Since these pins are alternate function pins in the general I/O ports, the corresponding pins must be set as input pins in the data direction register - DDRX.

The external interrupts are set up as indicated in the specification for the general interrupt mask register - GIMSK.

INTERRUPT RESPONSE TIME

The interrupt response time for all the enabled *AVR* interrupts are 4 clock cycles. After the 4 clock cycles the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2.

A return from an interrupt handling routine takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2.

Note that the Status Register - SREG - is not handled by the *AVR* hardware, neither for interrupts nor for subroutines. For the routines requiring a storage of the SREG, this must be performed by user software.

THE MCU CONTROL REGISTER - MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
$35	-	-	SE	SM	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7, 6 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read as zero.

Bit 5 - SE : Sleep Enable:
The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

Bit 4 - SM : Sleep Mode:
This bit selects between the two available sleep modes

When cleared (zero) the SM control bit forces the MCU into the Idle Mode stopping the CPU but allowing the General Purpose Working Registers, SRAM, Timer/Counters, SPI port, and interrupt system to continue operating.

When set (one), the SM control bit forces the MCU into the Power Down Mode saving the General Purpose Working Registers and the SRAM data, but freezing the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

To enter the sleep modes, the SE bit must be set (one) and a SLEEP instruction must be executed. The instruction following SLEEP is executed before entering sleep mode. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine and resumes execution from the address following the last executed instruction. If reset occurs while the MCU is in Sleep Mode, the MCU awakes and executes from the reset vector.

Bits 3, 2 - ISC11, ISC10 : Interrupt Sense Control 1 bit 1 and bit 0:
The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask in the GIMSK register is set. The level and edges on the external INT1 pin that activate the interrupt are defined in the following table:

Table 3: Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Note: When changing the ISC11/ISC10 bits, INT1 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bits 1, 0 - ISC01, ISC00 : Interrupt Sense Control 0 bit 1 and bit 0:

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask is set. The level and edges on the external INT0 pin that activate the interrupt are defined in the following table:

Table 4: Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Note: When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Timer / Counters

The AT90S2312 provides two general purpose Timer/Counters - one 8-bit T/C and one 16-bit T/C. The Timer/Counters have separate prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

The Timer/Counter Prescaler

Figure 21 shows the general Timer/Counter prescaler.

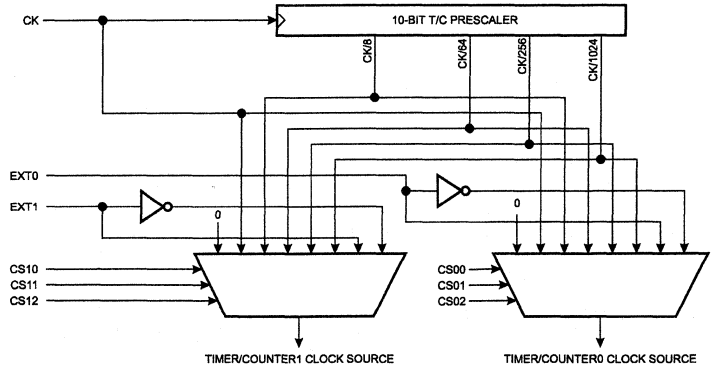


Figure 21: Timer/Counter Prescaler

The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK is the oscillator clock. For the two Timer/Counters, added selections as CK, external source and stop, can be selected as clock sources.

The 8-Bit Timer/Counter0

Figure 22 shows the block diagram for Timer/Counter0.

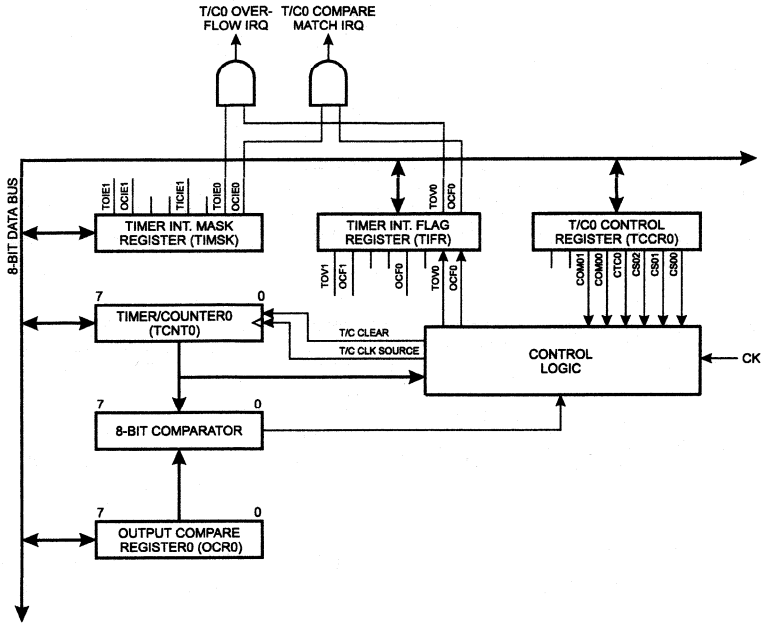


Figure 22: Timer/Counter 0 Block Diagram

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter0 Control Register - TCCR0. The different status flags (overflow and compare match) and control signals are found in the Timer/Counter0 Control Register - TCCR0. The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time for the external clock being low and high must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter0 supports an Output Compare function using the Output Compare Register 0 - OCR0 as the data source to be compared to the Timer/Counter0 contents. The Output Compare functions include optional clearing of the counter on compare matches, and actions on the Output Compare pin 0 on compare matches. The Output Compare pin 0 function makes the Timer/Counter0 useful for PWM (Pulse Width Modulation) functions.

THE TIMER/COUNTER0 CONTROL REGISTER - TCCR0

Bit	7	6	5	4	3	2	1	0	
$33	-	-	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7,6 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

Bits 5,4 - COM01, COM00 : Compare Output Mode0, bit 1 and 0:
The COM01 and COM00 control bits determine any output pin action following a compare match in Timer/Counter0. Any output pin actions are effective on pin OC0 - Output Compare pin 0. Since this is an alternative function to an I/O port, the corresponding data direction control bit must be set (one) to control an output pin. The control configuration is defined in the following table:

Table 5: Compare 0 Mode Select

COM01	COM00	Description
0	0	Timer/Counter0 disconnected from output pin OC0.
0	1	Toggle the OC0 output line.
1	0	Clear the OC0 output line (to zero).
1	1	Set the OC0 output line (to one).

Note: When changing the COM01/COM00 bits, Output Compare Interrupt 0 must be disabled by clearing its Interrupt Enable bit in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bit 3 - CTC0 : Clear Timer/Counter0 on Compare match:
When the CTC0 control bit is set (one), the Timer/Counter0 is reset to $00 in the clock cycle after the compare match. If the CTC0 control bit is cleared, the Timer/Counter0 continues running freely until it is stopped, set, cleared or wraps around (overflow).

Bits 2,1,0 - CS02, CS01, CS00 : Clock Select0, bit 2,1 and 0:
The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer0.

Table 6: Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T0, rising edge
1	1	1	External Pin T0, falling edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual data direction control register (cleared to zero gives an input pin).

Bits 5..3 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

THE TIMER COUNTER 0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
$32	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.

THE OUTPUT COMPARE REGISTER 0 - OCR0

Bit	7	6	5	4	3	2	1	0	
$31	MSB							LSB	OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Output Compare Register 0 is the source register for the Timer/Counter0 compare match functions.

The 16-Bit Timer/Counter1

Figure 23 shows the block diagram for Timer/Counter1.

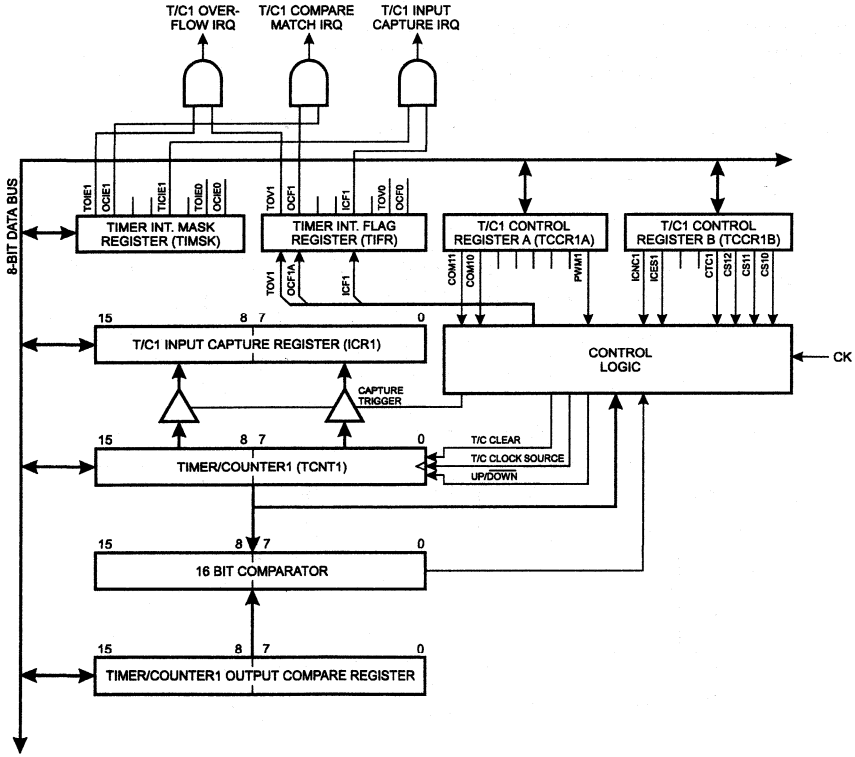


Figure 23: Timer/Counter1 Block Diagram

The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter1 Control Registers - TCCR1A and TCCR1B. The different status flags (overflow, compare match and capture event) and control signals are found in the Timer/Counter Interrupt Flag Register - TIFR. The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time for the external clock being low and high must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports an Output Compare function using the Output Compare Register 1 - OCR1 as the data source to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compare matches, and actions on the Output Compare pin 1 on compare matches.

Timer/Counter1 can also be used as a 10 bit Pulse With Modulator. In this mode the counter and the OCR1 register serve as a glitch-free stand-alone PWM with centered pulses. Refer to Page 3-31 for a detailed description on this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register - ICR1, triggered by an external event on the Input Capture Pin - ICP. The actual capture event settings are defined by the Timer/Counter1 Control Register - TCCR1.

The Timer/Counter1 input capture noise canceler block diagram is shown in Figure 24.

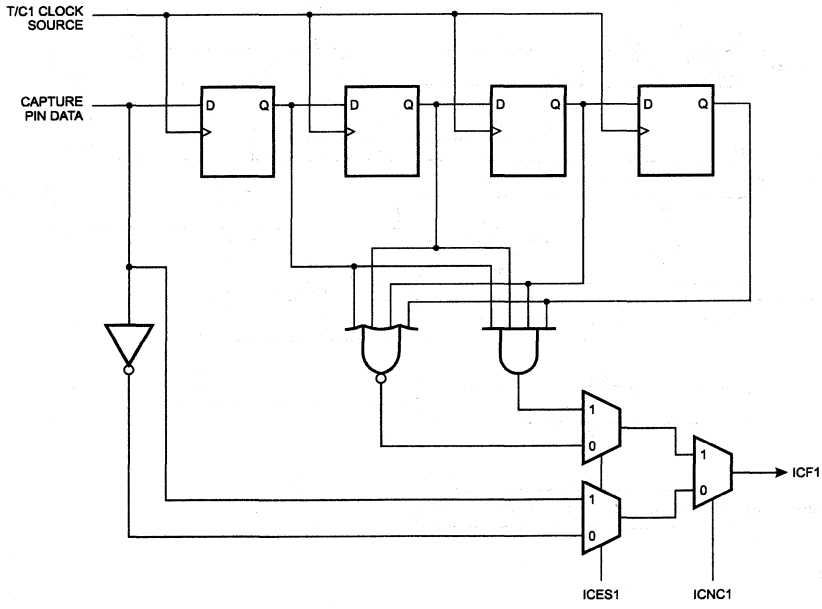


Figure 24: The Input Capture Noise Canceler

If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples before the capture is activated. The sampling clock is the same clock as the clock source selected for the Timer/Counter1.

THE TIMER/COUNTER1 CONTROL REGISTER A - TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F	COM11	COM10	-	-	-	-	-	PWM1	TCCR1A
Read/Write	R/W	R/W	R	R	R	R	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7,6 - COM11, COM10 : Compare Output Mode1, bits 1 and 0:
The COM11 and COM10 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1 - Output Compare pin 1. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 7.

Table 7: Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Notes: X = A or B
In PWM mode, these bits have a different function. Refer to Table 9 for a detailed description.
When changing the COM1X1/COM1X0 bits, Output Compare Interrupts 1 must be disabled by clearing their Interrupt Enable bits in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bits 5..1 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

Bit 0 - PWM1 : Pulse Width Modulator enable:
This bit enables the PWM mode for Timer/Counter1. This mode is described on Page 4-31.

THE TIMER/COUNTER1 CONTROL REGISTER B - TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	TCSR1B
Read/Write	R/W	R/W	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - ICNC1 : Input Capture1 Noise Canceler (4 CKs):
When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP - input capture pin - as specified. When the ICNC1 bit is set (one), four successive samples are measures on the ICP - input capture pin, and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is the same as the clock source frequency selected for the Timer/Counter1.

Bit 6 - ICES1 : Input Capture1 Edge Select:
While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the falling edge of the input capture pin - ICP. While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the rising edge of the input capture pin - ICP.

Bits 5, 4 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and always read zero.

Bit 3 - CTC1 : Clear Timer/Counter1 on Compare match:
When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000 in the clock cycle after a compare match. If the CTC1 control bit is cleared, the Timer/Counter1 continues counting until it is stopped, cleared, wraps around (overflow) or changes direction. In PWM mode, this bit has no effect.

Bits 2,1,0 - CS12, CS11, CS10 : Clock Select1, bit 2,1 and 0:
The Clock Select1 bits 2,1 and 0 define the prescaling source of Timer/Counter1.

Table 8: Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T1, rising edge
1	1	1	External Pin T1, falling edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual direction control register (cleared to zero gives an input pin).

THE TIMER/COUNTER1 - TCNT1H AND TCNT1L

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

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP).

- **TCNT1 Timer/Counter1 Write:**
  When the CPU writes to the low byte TCNT1L, the written data is placed in the TEMP register. Next, when the CPU writes the high byte TCNT1H, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written in the TCNT1 Timer/Counter1 register simultaneously. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register write operation.

- **TCNT1 Timer/Counter1 Read:**
  When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the clock cycle after it is preset with the written value.

TIMER/COUNTER1 OUTPUT COMPARE REGISTER - OCR1H AND OCR1L

Bit	15	14	13	12	11	10	9	8	
$2B	MSB								OCR1H
$2A								LSB	OCR1L
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The output compare register is a 16-bit read/write register.

The Timer/Counter1 Output Compare Register contains the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status register.

Since the Output Compare Register - OCR1 - is a 16-bit register, a temporary register TEMP is used when OCR1 is written to ensure that both bytes are updated simultaneously. When the CPU writes the low byte, OCR1L, the data is temporarily stored in the TEMP register. When the CPU writes the high byte, OCR1H, the TEMP register is simultaneously written to OCR1L. Consequently, the low byte OCR1L must be written first for a full 16-bit register write operation.

THE TIMER/COUNTER1 INPUT CAPTURE REGISTER - ICR1H AND ICR1L

Bit	15	14	13	12	11	10	9	8	
$25	MSB								ICR1H
$24								LSB	ICR1L
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting - ICES1) of the signal at the input capture pin - ICP - is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At the same time, the input capture flag - ICF1 - is set (one).

Since the Input Capture Register - ICR1 - is a 16-bit register, a temporary register TEMP is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU and the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, the CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit register read operation.

TIMER/COUNTER1 IN PWM MODE

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1 - OCR1, form a 10-bit, free-running, glitch-free and phase correct PWM with output on the PD1(OC1) pin. Timer/Counter1 acts as an up/down counter, counting up from $0000 to $03FF (i.e. from 0 to 1023), when it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 10 least significant bits of OCR1, the PD1(OC1)

pin is set or cleared according to the settings of the COM11 and COM10 bits in the Timer/Counter1 Control Register TCCR1. Refer to Table 9 for details.

Table 9: Compare1 Mode Select in PWM Mode

COM11	COM10	Effect on OC1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, upcounting. Set on compare match, downcounting (non-inverted PWM).
1	1	Cleared on compare match, downcounting. Set on compare match, upcounting (inverted PWM).

Note that in the PWM mode, the 10 least significant OCR1 bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the top - $03FF. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1 write. See Figure 25 for an example.

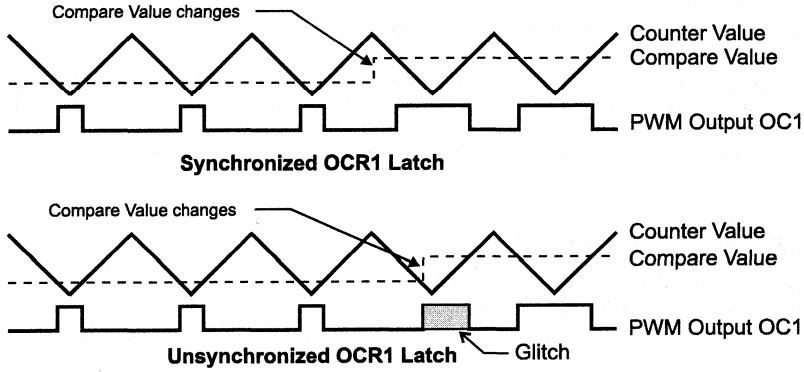


Figure 25: Effects on Unsynchronized OCR1 Latching

When OCR1 contains $0000 or $03FF, the output OC1 is held low or high according to the settings of COM11 and COM10. This is shown in Table 10.

Table 10: PWM Outputs OCR = $0000 or $03FF

COM11	COM10	OCR1	Output OC1
1	0	$0000	L
1	0	$03FF	H
1	1	$0000	H
1	1	$03FF	L

In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter changes direction at $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This does also apply to the Timer Output Compare1 flag and interrupt.

The PWM output frequency is given by: $f_{PWM} = \frac{f_{TC1}}{2046}$, where f_{TC1} is the Timer/Counter1 Clock Source frequency.

The Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted from 16 to 2048 ms. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. From the Watchdog is reset, eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S2312 resets and executes from the reset vector.

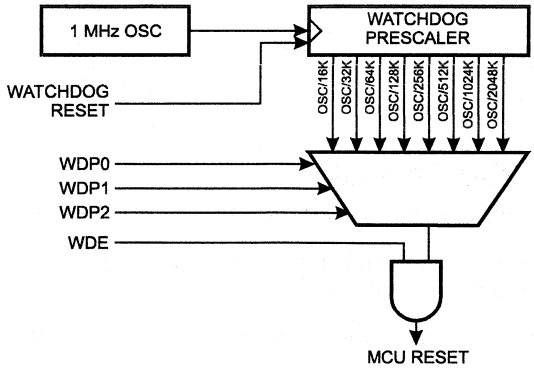


Figure 26: Watchdog Timer

THE WATCHDOG TIMER CONTROL REGISTER - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21	-	-	-	-	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..4 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and will always read as zero.

Bit 3 - WDE : Watch Dog Enable:
When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled.

Bits 2..0 - WDP2, WDP1, WDP0 : Watchdog Timer Prescaler 1 and 0:
The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding timeout periods are shown in Table 11.

Table 11: Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Timeout Period
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space using the IN and OUT instructions.

The write access time is in the range of 2.5 - 4ms, depending on the Vcc voltages. A self-timing function, however, lets the user software detect when the next byte can be written.

The read access time is the same as for the Flash memory and is of no concern to the user software.

THE EEPROM ADDRESS REGISTER - EEAR

Bit	7	6	5	4	3	2	1	0	
$1E	-	MSB						LSB	EEAR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - Res : Reserved bit:
This bit is a reserved bit in the AT90S2312 and will always read as zero.

Bit 6..0 - EEAR6..0 : EEPROM Address:

The EEPROM Address Register - EEAR6..0 - specifies the EEPROM address in the 128 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 127.

THE EEPROM DATA REGISTER - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7..0 - EEDR7..0 : EEPROM Data:
For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

THE EEPROM CONTROL REGISTER - EECR

Bit	7	6	5	4	3	2	1	0	
$1C	-	-	-	-	-	-	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7..2 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and will always read as zero.

Bit 1 - EEWE : EEPROM Write Enable:
The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. When the write access time (typically 2.5ms at Vcc=5V or 4ms at Vcc=2.7V) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte.

Bit 0 - EERE : EEPROM Read Enable:
The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access time is within a single clock cycle and there is no need to poll the EERE bit.

The UART

The AT90S2312 features a full duplex Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- Baud rate generator generates any baud rate
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data Transmission

A block schematic of the UART transmitter is shown in Figure 27.

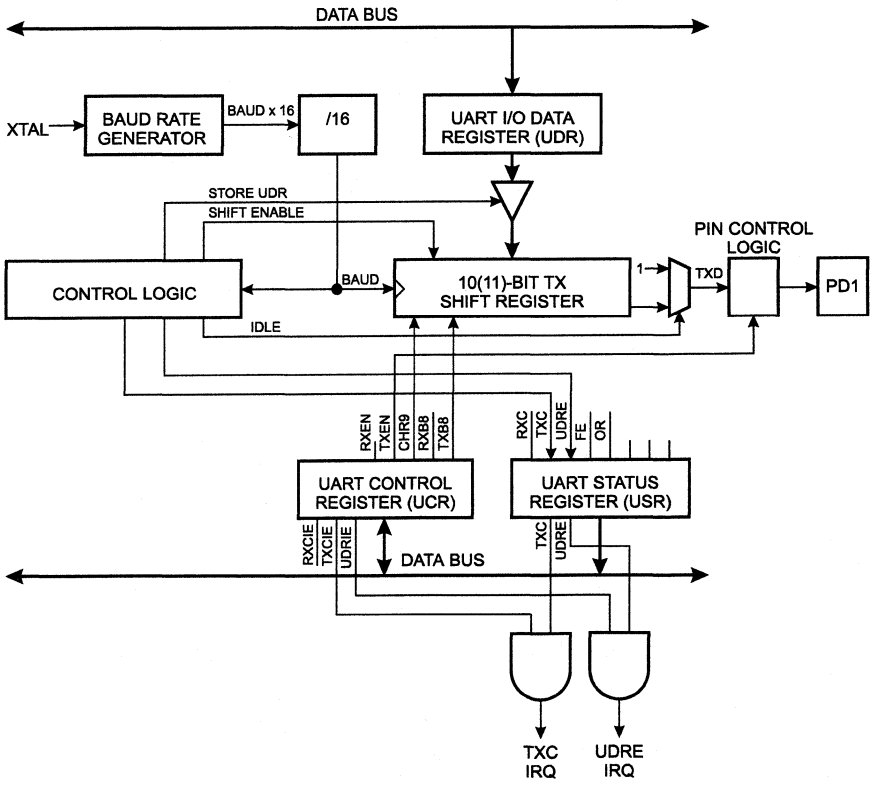


Figure 27: UART Transmitter

Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferred from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The shift register is loaded when the stop bit of the character currently being transmitted has been shifted out.

At this time the UDRE (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time as the data is transferred from UDR to the 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Transmit shift register.

On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to send when the stop bit is shifted out, the UDRE flag will remain set. In this case, after the stop bit has been present on TXD for one bit length, the TX Complete Flag, TXC, in USR is set.

The TXEN bit in UCR enables the UART transmitter when set (one). By clearing this bit (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to the PD1 pin regardless of the setting of the DDD1 bit in DDRD.

Data Reception

Figure 28 shows a block diagram of the UART Receiver

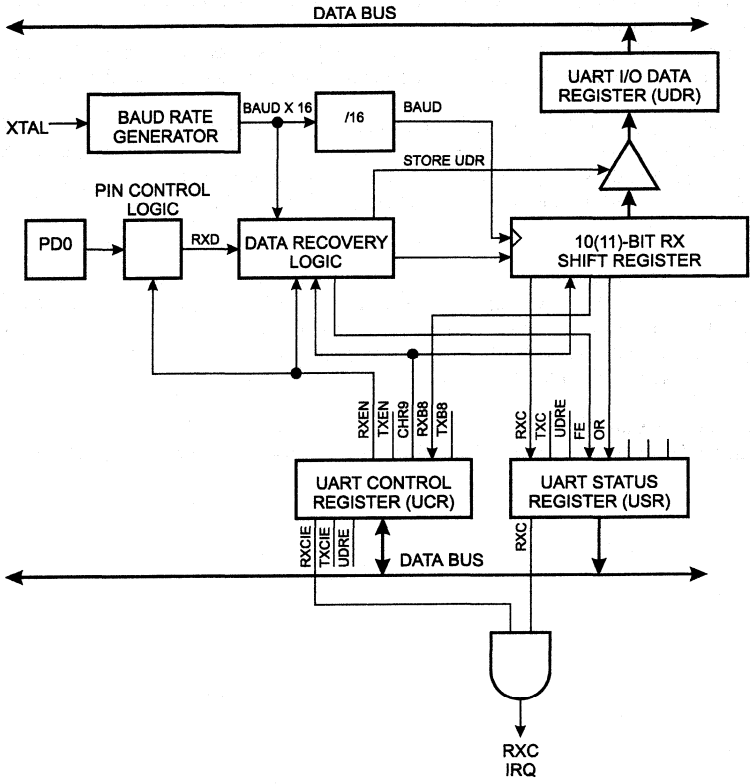


Figure 28: UART Receiver

The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1 to 0-transition, the receiver samples the RXD pin at sample 8, 9 and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0-transition.

If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the transmitter shift register as they are sampled. Sampling of an incoming character is shown in Figure 29.

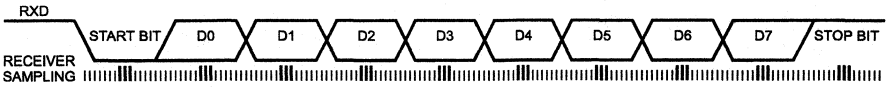


Figure 29: Sampling Received Data

When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logical zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been accessed since the last receive, the OverRun (OR) flag in UCR is set. This means that the new data transferred to the shift register has overwritten the old data not yet read, and the old data is lost. The user should always check the OR bit before reading from the UDR register in order to detect any overruns.

By clearing the RXEN bit in the UCR register, the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to the PD0 pin regardless of the setting of the DDD0 bit in DDRD.

UART Control

THE UART I/O DATA REGISTER - UDR

Bit	7	6	5	4	3	2	1	0	
$0C	MSB							LSB	UDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

THE UART STATUS REGISTER - USR

Bit	7	6	5	4	3	2	1	0	
$0B	RXC	TXC	UDRE	FE	OR	-	-	-	USR
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	1	1	0	0	0	0	0	

The USR register is a read-only register providing information on the UART Status.

Bit 7 - RXC: UART Receive Complete:
This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCR is set, setting of RXC causes the UART Receive Complete interrupt to be executed. RXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the bit is cleared (zero) by first reading USR while RXC is set (one) and then reading UDR.

Bit 6 - TXC : UART Transmit Complete:
This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out and no new data has been written to the UDR. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by first reading USR while TXC is set and then writing UDR.

This bit is set (one) during reset to indicate that the transmitter is not busy transmitting anything.

Bit 5 - UDRE : UART Data Register Empty:
This bit is set (one) when a character written to UDR is transferred to the Transmit shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, setting of UDRE causes the UART Transmit Complete interrupt to be executed. UDRE is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the UDRE bit is cleared (zero) by first reading USR while UDRE is set and then writing UDR.

UDRE is set (one) during reset to indicate that the transmitter is ready.

Bit 4 - FE : Framing Error:
This bit is set if a Framing Error condition is detected, i.e. when the stop bit of an incoming character is zero.

The FE bit is cleared (zero) by first reading USR while FE is set and then reading UDR.

Bit 3 - OR : OverRun:
This bit is set if an Overrun condition is detected, i.e. when a character already present in the UDR register is not read before the next character is transferred from the Receiver Shift register.

The OR bit is cleared (zero) by first reading USR while OR is set and then reading UDR.

Bits 2..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and will always read as zero.

THE UART CONTROL REGISTER - UCR

Bit	7	6	5	4	3	2	1	0	
$0A	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - RXCIE : RX Complete Interrupt Enable:
When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 6 - TXCIE : TX Complete Interrupt Enable:
When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 5 - UDRIE : UART Data Register Empty Interrupt Enable:
When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty interrupt routine to be executed provided that global interrupts are enabled.

Bit 4 - RXEN : Receiver Enable:
This bit enables the UART receiver when set (one). When the receiver is disabled, the TXC, OR and FE status flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

Bit 3 - TXEN : Transmitter Enable:
This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

Bit 2 - CHR9 : 9 Bit Characters: When this bit is set (one) transmitted and received characters are 9 bit long plus start and stop bits. The 9^{th} bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The 9^{th} data bit can be used as an extra stop bit or a parity bit.

Bit 1 - RXB8 : Receive Data Bit 8
When CHR9 is set (one), RXB8 is the 9^{th} data bit of the received character.

Bit 0 - TXB8 : Transmit Data Bit 8
When CHR9 is set (one), TXB8 is the 9^{th} data bit in the character to be transmitted.

THE BAUD RATE GENERATOR

The baud rate generator is a frequency divider which generates baud-rates according to the following equation:

$$BAUD = \frac{f_{ck}}{16(UBRR+1)}$$

- BAUD = Baud-Rate
- f_{ck} = Crystal Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0-255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings in Table 12. UBRR values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table.

Table 12: UBRR Settings at Various Crystal Frequencies

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBRR=	25	0.2	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	63	0.0
4800	UBRR=	12	0.2	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	31	0.0
9600	UBRR=	6	7.5	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	15	0.0
14400	UBRR=	3	7.8	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	0.0
28800	UBRR=	1	7.8	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	6.3
57600	UBRR=	0	7.8	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	12.5
115200	UBRR=	0	84.3	UBRR=	0	0.0	UBRR=	0	7.8	UBRR=	0	25.0

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBRR=	84	0.4	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0
4800	UBRR=	42	0.8	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0
9600	UBRR=	20	1.6	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0
14400	UBRR=	13	1.6	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0
19200	UBRR=	10	3.1	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0
28800	UBRR=	6	1.6	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0
57600	UBRR=	3	12.5	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0
115200	UBRR=	1	12.5	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	287	-
4800	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	143	0.0
9600	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	71	0.0
14400	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	47	0.0
19200	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0	UBRR=	35	0.0
28800	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	23	0.0
57600	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	11	0.0
115200	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0	UBRR=	5	0.0

Baud Rate	14.746 MHz		%Error	16 MHz		%Error	18.432 MHz		%Error	20 MHz		%Error
2400	UBRR=	383	-	UBRR=	416	-	UBRR=	479	-	UBRR=	520	-
4800	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	259	-
9600	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	129	0.2
14400	UBRR=	63	0.0	UBRR=	68	0.6	UBRR=	79	0.0	UBRR=	86	0.2
19200	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	64	0.2
28800	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	42	0.9
57600	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	21	1.4
115200	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	10	1.4

THE UART BAUD RATE REGISTER - UBRR

Bit	7	6	5	4	3	2	1	0	
$09	MSB							LSB	UBRR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	1	1	

The UBRR register is an 8-bit read/write register which specifies the UART Baud Rate according to the description on Page 3-39.

The Analog Comparator

The analog comparator compares the input values on the positive pin AIN0 (PB0) and the negative pin PB1(AIN1). When the voltage on the positive pin PB0 (AIN0) is higher than the voltage on the negative PB1 (AIN1), the Analog Comparator Output, ACO is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 30.

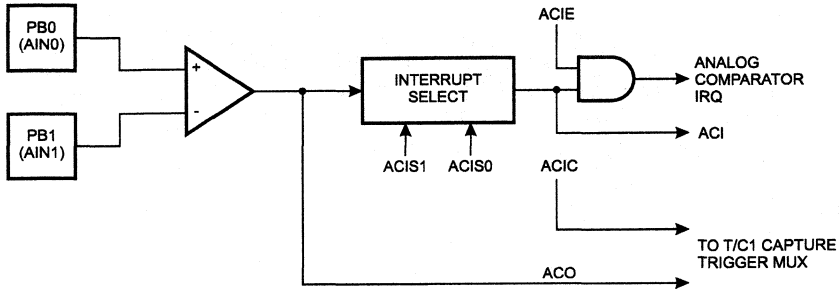


Figure 30: Analog Comparator Block Diagram

THE ANALOG COMPARATOR CONTROL AND STATUS REGISTER - ACSR

Bit	7	6	5	4	3	2	1	0	
$08	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..6 - Res : Reserved bits:
These bits are reserved bits in the AT90S2312 and will always read as zero.

Bit 5 - ACO : Analog Comparator Output:
ACO is directly connected to the comparator output.

Bit 4 - ACI : Analog Comparator Interrupt Flag:
This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

Bit 3 - ACIE : Analog Comparator Interrupt Enable:
When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled. For details on the comparator, refer to Page 4-43.

Bit 2 - ACIC : Analog Comparator Input Capture enable:
When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator

utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the analog comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

Bits 1,0 - ACIS1, ACIS0 : Analog Comparator Interrupt Mode Select:
These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in Table 13.

Table 13: ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

I/O-Ports

Port B

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

Three data memory address locations are allocated for the Port B, one each for the Data Register - PORTB ($18), Data Direction Register - DDRB ($17) and the Port B Input Pins - PINB ($16). The Port B Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pullups. The Port B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

The Port B pins with alternate functions are shown in the following table:

Table 14: Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	AIN0 (Analog comparator positive input)
PB1	AIN1 (Analog comparator negative input)
PB2	OC0 (Timer/Counter0 Output compare match output)
PB3	OC1 (Timer/Counter1 Output compare match output)
PB5	MOSI (Data input line for memory downloading)
PB6	MISO (Data output line for memory uploading)
PB7	SCK (Master clock input)

When the pins are used for the alternate function the DDRB and PORTB register has to be set according to the alternate function description.

THE PORT B DATA REGISTER - PORTB

Bit	7	6	5	4	3	2	1	0	
$18	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B DATA DIRECTION REGISTER - DDRB

Bit	7	6	5	4	3	2	1	0	
$17	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B INPUT PINS ADDRESS - PINB

Bit	7	6	5	4	3	2	1	0	
$16	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the PORTB Data Latch is read, and when reading PINB, the logical values present on the pins are read.

PORTB AS GENERAL DIGITAL I/O

All 8 bits in port B are equal when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PBn is set (one) when the pin configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PBn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 15: DDBn Effects on Port B Pins

DDBn	PBn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PBn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTB

The alternate pin functions of Port B are:

SCK - PORTB, Bit 7:

SCK , Clock input pin for Memory downloading.

MISO - PORTB, Bit 6:

MISO, Data output pin for Memory downloading.

MOSI - PORTB, Bit 5:

MOSI , Data input pin for Memory downloading.

OC1 - PORTB, Bit 3:

OC1, Output compare match output: The PB3 pin can serve as an external output when timer 1 compare match. The PB3 pin has to be configured as an output (DDB3 is set (one)) to serve this function. See the timer description for further details, and how to enable the output.

OC0 - PORTB, Bit 2:

OC0, Output compare match output: The PB2 pin can serve as an external output when timer 0 compare match. The PB2 pin has to be configured as an output (DDB2 is set (one)) to serve this function. See the timer description for further details, and how to enable the output.

AIN1 - PORTB, Bit 1:

AIN1, Analog Comparator Negative Input. When configured as an input (DDB1 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB1 is cleared (zero)), this pin also serves as the negative input of the on-chip analog comparator.

AIN0 - PORTB, Bit 0:

AIN0, Analog Comparator Positive Input. When configured as an input (DDB0 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB0 is cleared (zero)), this pin also serves as the positive input of the on-chip analog comparator.

PORTB SCHEMATICS

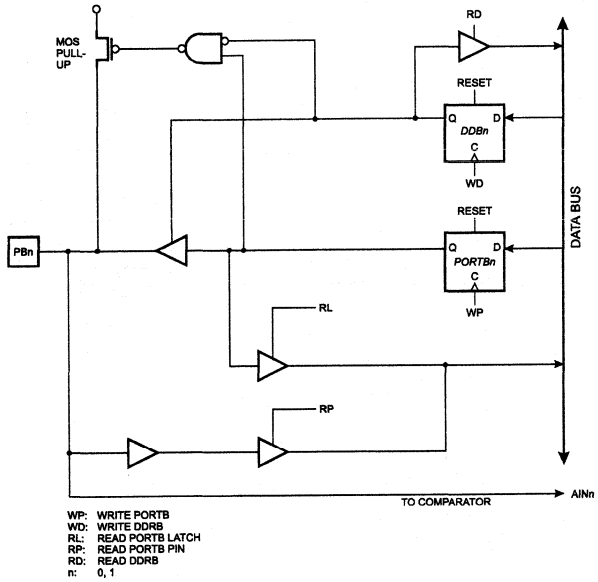


Figure 31: PORTB Schematic Diagram (pins PB0 and PB1)

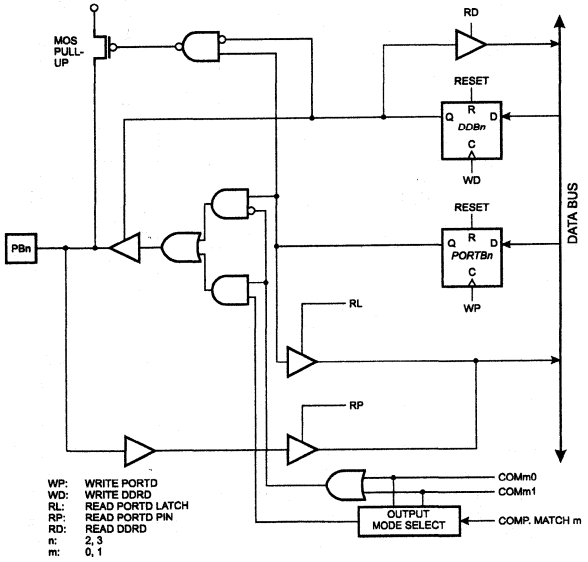


Figure 32: PORTB Schematic Diagram (Pins PB2 and PB3)

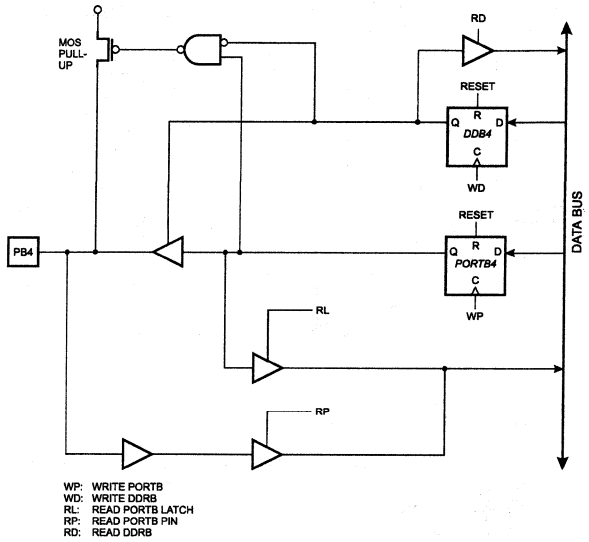


Figure 33: PORTB Schematic Diagram, Pin PB4

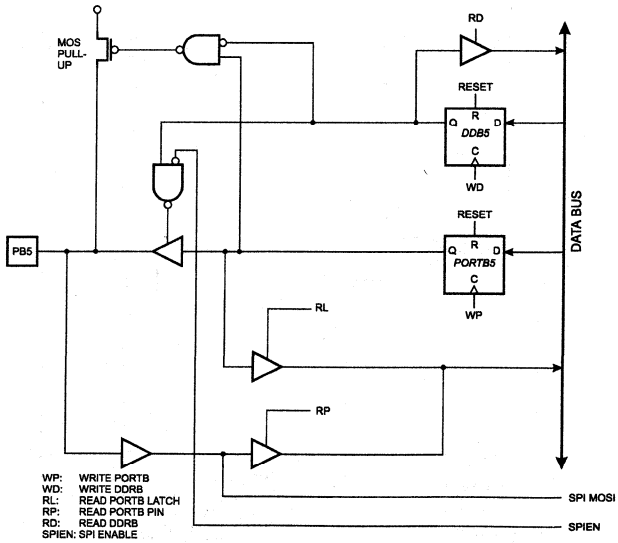


Figure 34: PORTB Schematic Diagram Pin PB5

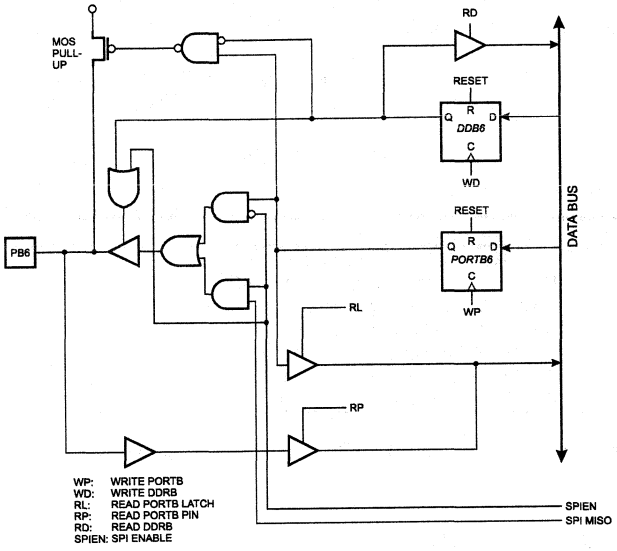


Figure 35: PORTB Schematic Diagram, Pin PB6

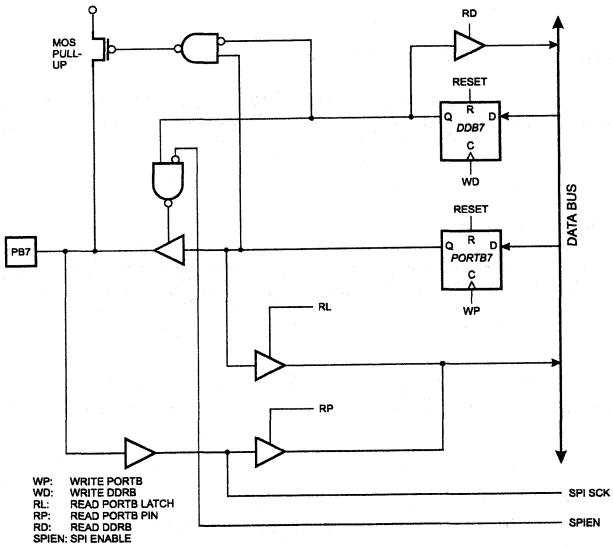


Figure 36: PORTB Schematic Diagram, Pin PB7

Port D

Three data memory address locations are allocated for the Port D, one each for the Data Register - PORTD ($12), Data Direction Register - DDRD ($11) and the Port D Input Pins - PIND ($10). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

Port D has seven bi-directional I/O pins with internal pullups, PD6..PD0. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Some Port D pins have alternate functions as shown in the following table:

Table 16: Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RXD (Receive data input for the UART)
PD1	TXD (Transmit data output for the UART)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD4	TO (Timer/Counter0 external input)
PD5	T1 (Timer/Counter1 external input)
PD6	ICP (Timer/Counter1Input Capture pin)

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alternate function description.

THE PORT D DATA REGISTER - PORTD

Bit	7	6	5	4	3	2	1	0	
$12	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0	PORTD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D DATA DIRECTION REGISTER - DDRB

Bit	7	6	5	4	3	2	1	0	
$11	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D INPUT PINS ADDRESS

Bit	7	6	5	4	3	2	1	0	
$10	-	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the PORTD Data Latch is read, and when reading PIND, the logical values present on the pins are read.

PORTD AS GENERAL DIGITAL I/O

PDn, General I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when

configured as an input pin the MOS pull up resistor is activated. To switch the pull up resistor off the PDn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 17: DDDn Bits on Port D Pins

DDDn	PDn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTD

The alternate functions of Port D are:

ICP - PORTD, Bit 6:

Timer/Counter1 Input Capture pin. The PD6 pin has to be configured as an input (DDD6 cleared (zero)) to serve this function. See the Timer/Counter1 description for further details. The internal pull up MOS resistor can be activated as described above.

T1 - PORTD, Bit 5:

T1, :Timer 1 clock source. The PD5 pin has to be configured as an input (DDD5 is cleared (zero)) to serve this function. See the timer description for further details. The internal pull up MOS resistor can be activated as described above.

T0 - PORTD, Bit 4:

T0, Timer/Counter0 clock source: The PD4 pin has to be configured as an input (DDD4 is cleared (zero)) to serve this function. See the Timer description for further details. The internal pull up MOS resistor can be activated as described above.

INT1 - PORTD, Bit 3:

INT1, External Interrupt source 1: The PD3 pin can serve as an external active low interrupt source to the MCU. The PD3 pin has to be configured as an input (DDD3 is cleared (zero)) to serve this function. The internal pull up MOS resistor can be activated as described above. See the interrupt description for further details, and how to enable the source.

INT0 - PORTD, Bit 2:

INT0, External Interrupt source 0: The PD2 pin can serve as an external active low interrupt source to the MCU. The PD2 pin has to be configured as an input (DDD2 is cleared (zero)) to serve this function. The internal pull up MOS resistor can be activated as described above. See the interrupt description for further details, and how to enable the source.

TXD - PORTD, Bit 1:

Transmit Data (Data output pin for the UART)

RXD - PORTD, Bit 0:

Receive Data (Data input pin for the UART)

PORTD SCHEMATICS

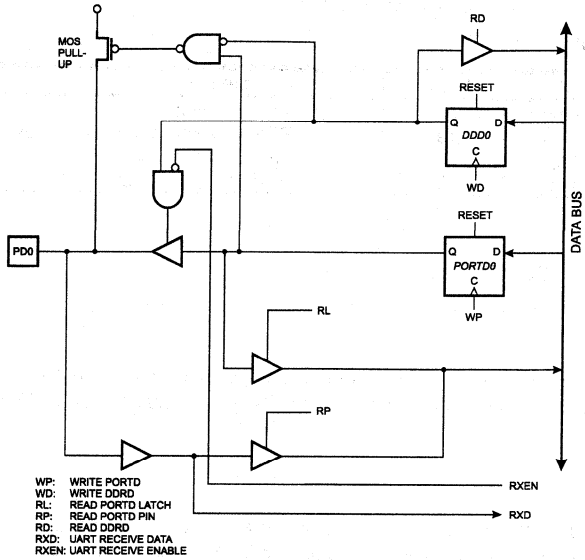


Figure 37: PORTD Schematic Diagram (Pin PD0)

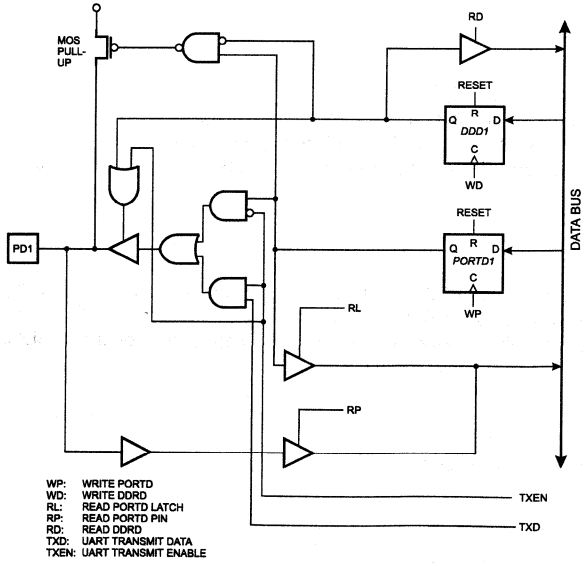


Figure 38: PORTD Schematic Diagram, Pin PD1

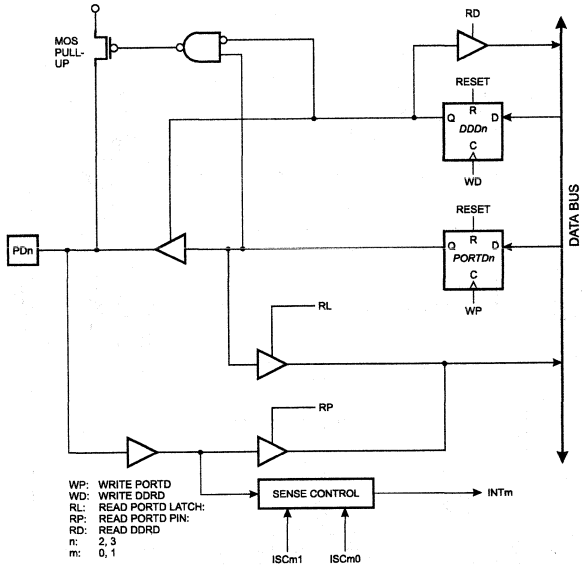


Figure 39: PORTD Schematic Diagram (Pins PD2 and PD3)

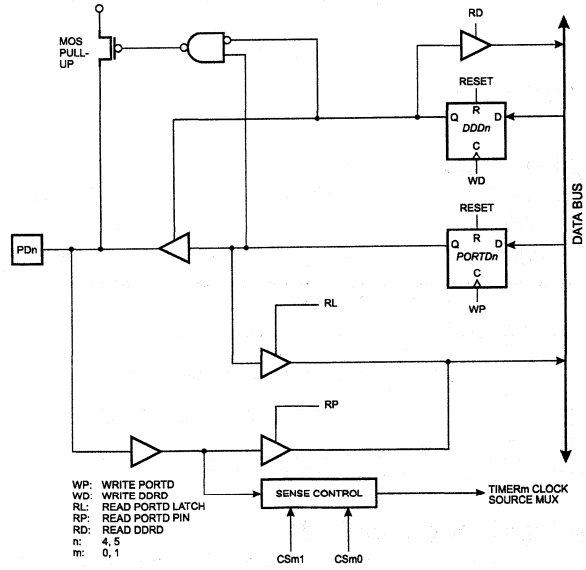


Figure 40: PORTD Schematic Diagram (Pins PD4 and PD5)

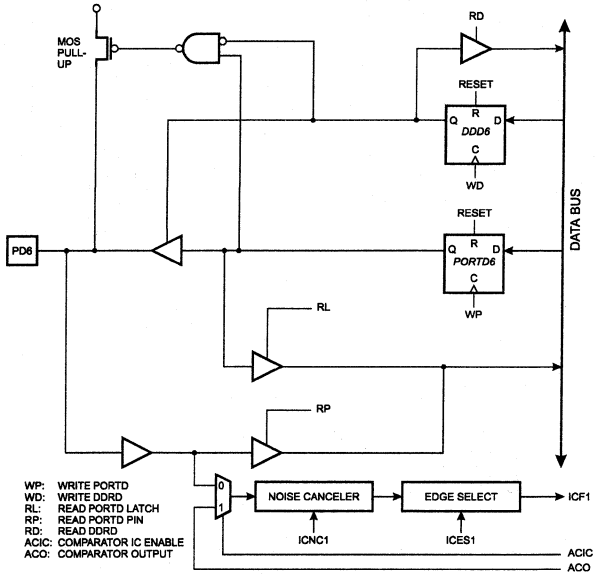


Figure 41: PORTD Schematic Diagram (Pin PD6)

Memory Programming

Program Memory Lock Bits

The AT90S2312 MCU provides two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 18.

Table 18: Lock Bit Protection Modes

Program Lock Bits			Protection Type
Mode	LB1	LB2	
1	U	U	No program lock features
2	P	U	Further programming of the Flash is disabled
3	P	P	Same as mode 2, but verify is also disabled.

Note: The Lock Bits can only be erased with the Chip Erase operation.

Programming the Flash and EEPROM

Atmel's AT90S2312 offers 2K bytes of in-system reprogrammable Flash PEROM Program memory and 128 bytes of EEPROM Data memory.

The AT90S2312 is normally shipped with the on-chip PEROM Program and EEPROM Data memory arrays in the erased state (i.e. contents = $FF) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The serial programming mode provides a convenient way to download the Program and Data into the AT90S2312 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Program and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one continuos address space: $000 to $7FF for the Program array and $800 to $87For the Data array.

The Program and Data memory arrays on the AT90S2312 are programmed byte-by-byte in either programming modes. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode. In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming

INTERNAL ADDRESS COUNTER

The AT90S2312 contains an internal Flash address counter which is always resets to $000 on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

PARALLEL PROGRAMMING ALGORITHM

All major programming vendors offer worldwide support for the Atmel microcontroller series. To program and verify the AT90S2312 in the parallel programming mode, the following sequence is recommended:

1. *Power-up sequence:*
   Apply power between VCC and GND pins. Set RST and XTAL1 to GND.
   With all other pins floating, wait for more than 10 ms.
2. Set RST to 'H'.
   Set PD2 to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins PD3, PD4, PD5, PD6 to select one of the programming operations shown in Table 19.
4. *Programming and verifying:*
   Apply data for Program memory location $000 to PB0 - PB7.
5. Raise RST to 12V to enable programming.
6. Pulse PD2 once to program a byte in the Program memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins PD3 to PD6 to the appropriate levels. Output data can be read at the port PB pins.
8. To program a byte at the next address location, pulse the XTAL1 pin once to advance the internal address counter. Apply new data to the port PB pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2KB array + 128 bytes of EEPROM or until the end of the object file is reached.
10. *Power-off sequence:*
    Set XTAL1 to 'L'.
    Set RST to 'L'.
    Float all other I/O pins.
    Turn VCC power off.

$\overline{\text{DATA}}$ POLLING

The AT90S2312 features $\overline{\text{DATA}}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel programming mode, an attempted read of the last byte written will result in the complement of the written datum on PB7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{DATA}}$ Polling may begin any time after a write cycle has been initiated.

READY/$\overline{\text{BUSY}}$

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin PD1 is pulled Low after PD2 goes High during programming to indicate $\overline{\text{BUSY}}$. PD1 is pulled High again when programming is done to indicate READY.

PROGRAM VERIFY

If lock bits LB1 and LB2 have not been programmed, Program data can be read back via the data lines for verification:

1. Reset the internal address counter to $000 by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Program data and read the output data at the port PB pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next Program data byte at the port PB pins.
5. Repeat steps 3 and 4 until the entire array is read. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

CHIP ERASE

The entire Flash array (2 Kbytes) + EEPROM (128 bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding PD2 low for 10 ms. The Program array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

READING THE SIGNATURE BYTES

The signature bytes are read by the same procedure as a normal verification of locations $000 and $001, except that PD5 and PD6 must be pulled to a logic low. The values returned are as follows.
($000) = $1E indicates manufactured by Atmel.
($001) = $91 indicates 2K bytes Program Flash Memory.
($002) = $01 indicates 90S2312 device when ($001) = $91.

Table 19: Flash and EEPROM Parallel Programming Modes

Mode	RST	PD2/ PROG	PD3	PD4	PD5	PD6	Data I/O PB7:0
Chip Erase	12V	(1)	H	L	L	L	X
Write Memory[2,3]	12V		L	H	H	H	DIN
Read Memory[2]	H	H	L	L	H	H	DOUT
Write Lock Bit 1	12V		H	H	H	H	X
Write Lock Bit 2	12V		H	H	L	L	X
Read Signature Byte	H	H	L	L	L	L	DOUT
Serial Prog. Enable	H	(1)	L	H	L	H	PB0=0
Serial Prog. Disable	H	(1)	L	H	L	H	PB0=1
Read Serial Prog. Fuse	H	L	H	H	L	H	PB0=SPF

Notes:

1. Chip erase and Serial Programming fuse requires a 10 ms $\overline{\text{PROG}}$ pulse.
2. The internal Flash address counter is reset to $000 on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.
3. PD1 is pulled Low during programming to indicate RDY/$\overline{\text{BSY}}$.

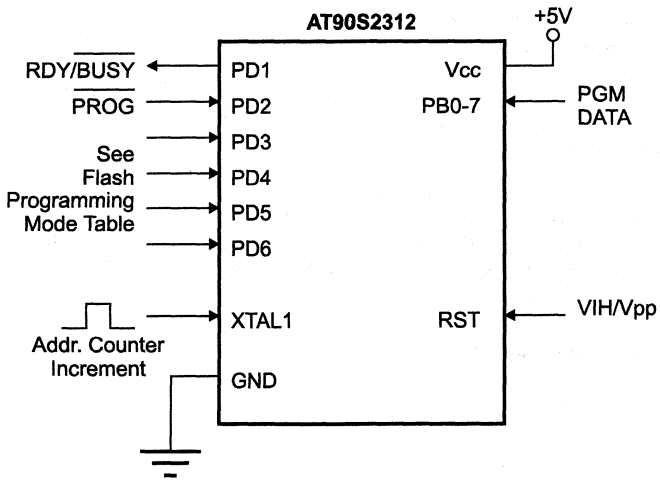


Figure 42: Parallel Programming

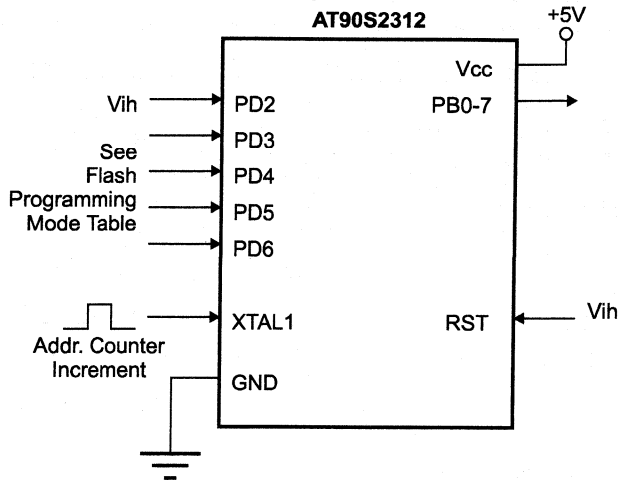


Figure 43: Parallel Verify

Serial Downloading

Both the Program and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to VCC. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and Data arrays into $FF.

The Program and Data memory arrays have separate address spaces:
$000 to $3FF for Program memory and $000 to $03F for Data memory.
Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 10 MHz oscillator clock, the maximum SCK frequency is 250 kHz.

SERIAL PROGRAMMING ALGORITHM

To program and verify the AT90S2312 in the serial programming mode, the following sequence is recommended (See three byte instruction formats in Table 20):

1. *Power-up sequence:*
   Apply power between VCC and GND.
   Set RST pin to 'H'.
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 1 MHz to 10 MHz clock to the XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB5. The frequency of the shift clock supplied at pin SCK/PB7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Program or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB6.
   $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
5. At the end of the programming session, RST can be set low to commence normal operation.
6. *Power-off sequence (if needed):*
   Set XTAL1 to 'L' (if a crystal is not used).
   Set RST to 'L'
   Turn Vcc power off.

Table 20: Serial Programming Instruction Set

Instruction	Instruction Format Byte 1	Byte 2	Byte3	Operation
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable Serial Programming after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 2K & 128 byte memory arrays
Read Program Memory	00aa a001	bbbb bbbb	oooo oooo	Read data **o** from Program memory at address **a:b**
Write Program Memory	00aa a010	bbbb bbbb	iiii iiii	Write data **i** to Program memory at address **a:b**
Read Data Memory	0000 0101	xbbb bbbb	oooo oooo	Read data **o** from Data memory at address **b**
Write Data Memory	0000 0110	xbbb bbbb	iiii iiii	Write data **i** to Data memory at address **b**
Write Lock Bits	1010 1100	0*12*x x111	xxxx xxxx	Write lock bits. Set bits ***1,2***='0' to program lock bits.

Note: **a** = address high bits
b = address low bits
o = data out
i = data in
x = don't care
1 = lock bit 1
2 = lock bit 2

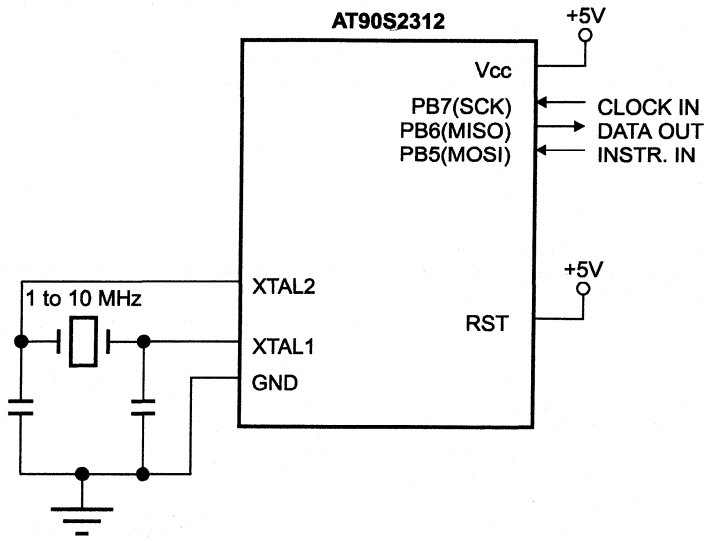


Figure 44: Serial Programming and Verify

When *writing* serial data to the AT90S2312, data is clocked on the *rising* edge of CLK.
When *reading* data from the AT90S2312, data is clocked on the *falling* edge of CLK. See Figure 46 for an explanation.

Programming Characteristics

Table 21: Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, Vcc = 5.0V±10%.

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold after $\overline{PROG}$	1.0		µs
t_{EHSH}	PD4 ($\overline{ENABLE}$)High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$ Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

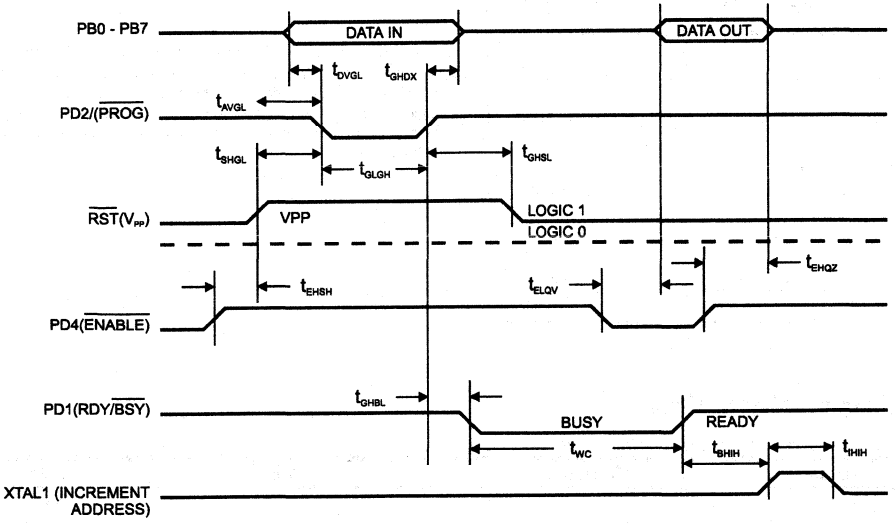


Figure 45: Flash/EEPROM Programming and Verfication Waveforms - Parallel Mode

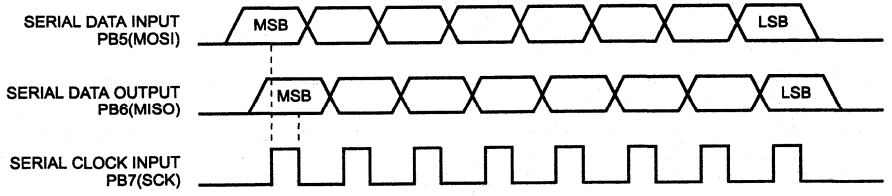


Figure 46: Serial Downloading Waveforms

Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin with respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	25.0 mA

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

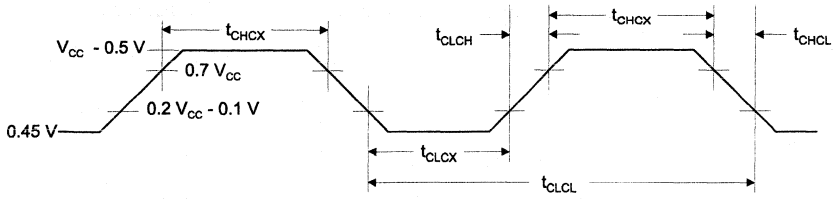
D.C. Characteristics

T_A=-40°C to 85°C Vcc=2.7V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		-0.5		0.2 Vcc - 0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RESET)	0.2 VCC + 0.9		VCC + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RESET)	0,7 VCC		VCC + 0.5	V
V_{OL}	Output Low Voltage[1] (Ports B, D)	I_{OL} = 20 mA, VCC = 5 V I_{OL} = 10 mA, VCC = 2.7 V			0.5	V
V_{OH}	Output High Voltage (Ports B,D)	I_{HI} = 10 mA, VCC = 5 V I_{HL} = 5 mA, VCC = 2.7 V	4.5			V
I_{OH}	Output Source Current (Ports B,D)	VCC = 5 V VCC = 2.7 V			10 5	mA
I_{OL}	Output Sink Current (Port B,D)	VCC = 5 V VCC = 2.7 V			20 10	mA
RRST	Reset Pulldown Resistor		10		50	KΩ
I_{CC}	Power Supply Current	Active Mode, 3V, 4MHz		2.5		mA
		Idle Mode 3V, 4MHz		800		uA
I_{CC}	Power Down Mode[2]	WDT enabled, 3V		50		µA
		WDT disabled, 3V		<1		µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 20mA
   Maximum total I_{OL} for all output pins: 80mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

External Clock Drive Waveforms



3

External Clock Drive

Symbol	Parameter	VCC = 2.7 V to 6.0 V		VCC = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	10	0	24	MHz
t_{CLCL}	Clock Period	100		41.7		ns
t_{CHCX}	High Time	40		16.7		ns
t_{CLCX}	Low Time	40		16.7		ns
t_{CLCH}	Rise Time		10		4.15	ns
t_{CHCL}	Fall Time		10		4.15	ns

Ordering Information

Ordering Code	Package	Operation Range
AT90S2312-PC AT90S2312-SC	20P3 20S	Commercial (0°C to 70°C)
AT90S2312-PI AT90S2312-SI	20P3 20S	Industrial (-40°C to 85°C)

Package Type	
20P3	20 Lead, 0.300" Wide Plastic Dual In-Line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

AT90S2312 Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F	SREG	I	T	O	S	V	N	Z	C	3-18
$3E	Reserved									
$3D	SPL	-	SP6	SP5	SP4	SP3	SP2	SP1	SP0	3-19
$3C	Reserved									
$3B	GIMSK	INT1	INT0	-	-	-	-	-	-	3-20
$3A	Reserved									
$39	TIMSK	TOIE1	OCIE1	-	-	TICIE1	-	TOIE0	OCIE0	3-21
$38	TIFR	TOV1	OCF1	-	-	ICF1	-	TOV0	OCF0	3-22
$37	Reserved									
$36	Reserved									
$35	MCUCR	-	-	SE	SM	ISC11	ISC10	ISC01	ISC00	3-23
$34	Reserved									
$33	TCCR0	TOV0	OCF0	COM01	COM00	CTC0	CS02	CS01	CS00	3-26
$32	TCNT0	Timer/Counter0 (8 Bit)								3-27
$31	OCR0	Timer/Counter0 Output Compare Register								3-27
$30	Reserved									
$2F	TCCR1A	COM11	COM10	-	-	-	-	-	PWM1	3-29
$2E	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	3-29
$2D	TCNT1H	Timer/Counter1 - Counter Register High Byte								3-29
$2C	TCNT1L	Timer/Counter1 - Counter Register Low Byte								3-29
$2B	OCR1H	Timer/Counter1 - Compare Register High Byte								3-31
$2A	OCR1L	Timer/Counter1 - Compare Register Low Byte								3-31
$29	Reserved									
$28	Reserved									
$27	Reserved									
$26	Reserved									
$25	ICR1H	Timer/Counter1 - Input Capture Register High Byte								3-31
$24	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								3-31
$23	Reserved									
$22	Reserved									
$21	WDTCR	-	-	-	-	WDE	WDP2	WDP1	WDP0	3-33
$20	Reserved									
$1F	Reserved									
$1E	EEAR	-	EEPROM Address Register							3-34
$1D	EEDR	EEPROM Data register								3-34
$1C	EECR	-	-	-	-	-	-	EEWE	EERE	3-35
$1B	Reserved									
$1A	Reserved									
$19	Reserved									
$18	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	3-44
$17	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	3-45
$16	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	3-45
$15	Reserved									
$14	Reserved									
$13	Reserved									
$12	PORTD	-	PD6	PD5	PD4	PD3	PD2	PD1	PD0	3-50
$11	DDRD	-	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	3-50
$10	PIND	-	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	3-50
$0F	Reserved									
$0E	Reserved									
$0D	Reserved									
$0C	UDR	UART I/O Data Register								3-39
$0B	USR	RXC	TXC	UDRE	FE	OR	-	-	-	3-39
$0A	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	3-40
$09	UBRR	UART Baud Rate Register								3-42
$08	ACSR	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	3-42
...	Reserved									
$00	Reserved									

AT90S2312 Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC + k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC + k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

(continued)

AT90S2312 Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1

Contents

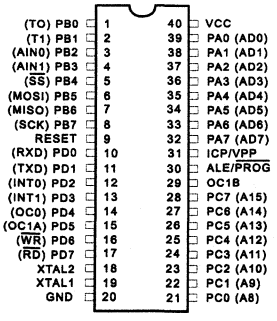
4

Features

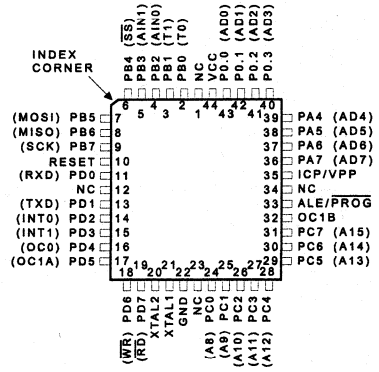
- Utilizes the AVR™ Enhanced RISC Architecture
- AVR™ - High Performance and Low Power RISC Architecture
- 112 Powerful Instructions - Most Single Clock Cycle Execution
- Efficient Context Switching Concept
- 8 Kbytes of In-System Reprogrammable Downloadable Flash
  SPI Serial Interface for Program Downloading
  Endurance: 1,000 Write/Erase Cycles
- 256 bytes EEPROM
  Endurance: 100,000 Write/Erase Cycles
- 256 bytes Internal RAM
- 32 x 8 General Purpose Working Registers
- 32 Programmable I/O Lines
- Programmable Serial UART
- SPI Serial Interface
- VCC Min.: 2.7 V
- Fully Static Operation
- One 8-Bit Timer/Counter with Separate Prescaler
- One 16-Bit Timer/Counter with Separate Prescaler and Compare and Capture Modes
- Dual 10 bit PWM
- External and Internal Interrupt Sources
- Programmable Watchdog Timer
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes
- Programming Lock for Software Security

Pin Configurations

PDIP



PLCC



8-Bit AVR Microcontroller with 8K bytes Downloadable Flash

4

Block Diagram

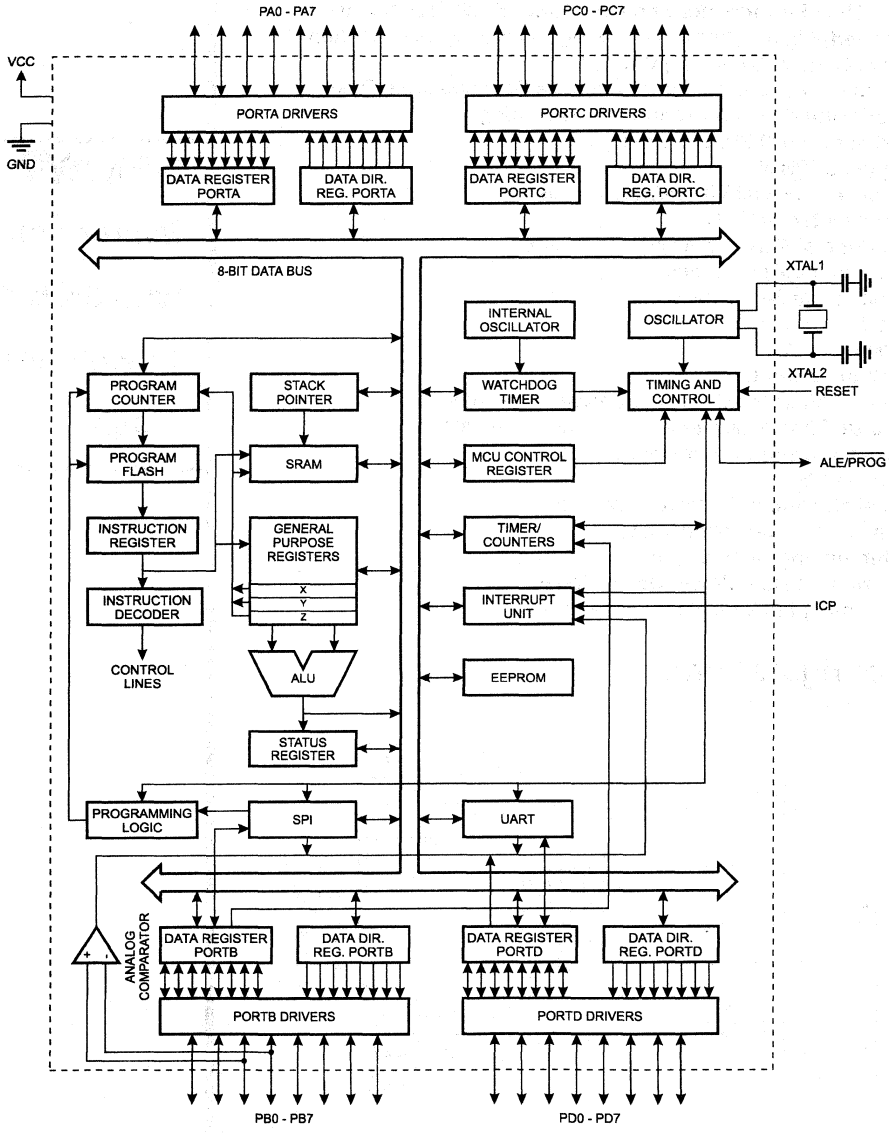


Figure 1: The AT90S8414 Block Diagram

Description

The AT90S8414 is a low-power CMOS 8 bit microcontroller based on the *AVR* enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the AT90S8414 achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

The *AVR* core is based on an enhanced RISC architecture that combines a rich instruction set with the 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (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 architecture supports high level languages efficiently while on-chip context switching hardware allows high performance, low power real timer control system design.

The AT90S8414 provides the following features: 8K bytes of Downloadable Flash, 256-bytes EEPROM, 256-bytes SRAM, 32 general purpose I/O lines, 32 general purpose working registers, flexible timer/counters with compare modes, internal and external interrupts, a programmable serial UART, programmable Watchdog Timer with internal oscillator, an SPI serial port and two software selectable power saving modes. The Idle Mode stops the CPU while allowing the SRAM, timer/counters, SPI port and interrupt system to continue functioning. The power down mode saves the register contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The device is manufactured using Atmel's high density non-volatile memory technology. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining an enhanced RISC 8 bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT90S8414 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The AT90S8414 *AVR* is supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

Pin Descriptions

VCC
Supply voltage

GND
Ground

Port A (PA7..PA0)
Port A is an 8-bit bidirectional I/O port. Port pins can provide internal pullups (selected for each bit). The Port A output buffers can sink 20mA and can drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

Port A serves as Multiplexed Address/Data input/output when using external SRAM.

Port B (PB7..PB0)
Port B is an 8-bit bidirectional I/O pins with internal pullups. The Port B output buffers can sink 20 mA. As inputs, Port B pins that are externally pulled low will source current (IIL) if the pullups are activated.

Port B also serves the functions of various special features of the AT90S8414 as listed on Page 4-58.

Port C (PC7..PC0)
Port C is an 8-bit bidirectional I/O port with internal pullups. The Port C output buffers can sink 20 mA. As inputs, Port C pins that are externally pulled low will source current (IIL) if the pullups are activated.

Port C also serves as Address output when using external SRAM.

Port D (PD7..PD0)
Port D is an 8-bit bidirectional I/O port with internal pullups. The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Port D also serves the functions of various special features of the AT90S8414 as listed on Page 4-65.

RESET
Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier

ICP / VPP
ICP is the input pin for the Timer/Counter1 Input Capture function.

When programming the AT90S8414 in parallel programming mode, the programming voltage, VPP is applied to this pin.

OC1B
OC1B is the output pin for the Timer/Counter1 Output CompareB function

ALE / $\overline{\text{PROG}}$
ALE is the Address Latch Enable used when the External Memory is enabled. The ALE strobe is used to latch the low-order address (8 bits) into an address latch during the first access cycle, and the AD0-7 pins are used for data during the second access cycle.

In parallel programming mode, this pin is pulsed to write data.

Crystal Oscillator

XTAL1 and XTAL2 are input and output, respectively , of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 2. Either a quartz crystal or a ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

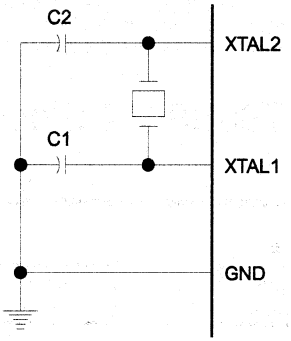


Figure 2: Oscillator Connections

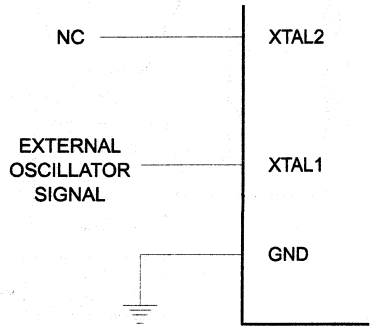


Figure 3: External Clock Drive Configuration

AT90S8414 *AVR* RISC Microcontroller CPU

The AT90S8414 *AVR* RISC microcontroller is upward compatible with the *AVR* Enhanced RISC Architecture. The programs written for the AT90S8414 MCU are fully compatible with the range of *AVR* 8-bit MCUs (AT90Sxxxx) with respect to source code and clock cycles for execution.

Architectural Overview

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 ALU (Arithmetic Logic Unit) 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-bits indirect address register pointers for SRAM addressing - enabling efficient address calculations. One of the three address pointers is also used as the address pointer for the constant table look up function. These added function registers are the 16-bits X-register, Y-register and Z-register.

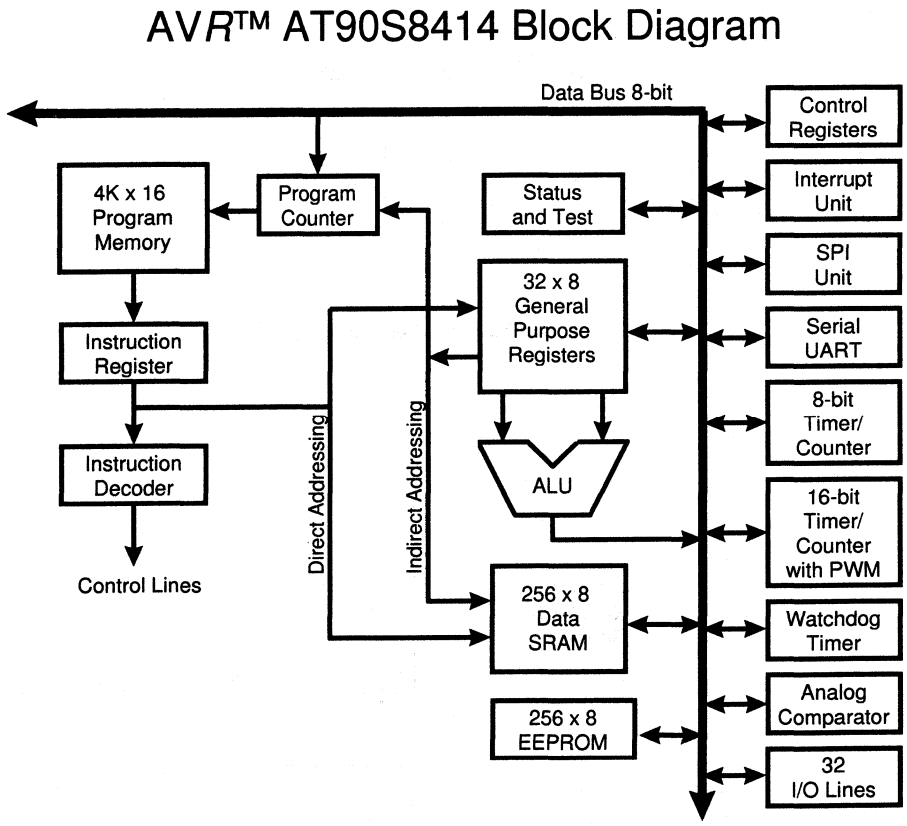


Figure 4: The AT90S8414 *AVR* Enhanced RISC Architecture

The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single register operations are also executed in the ALU. Figure 4 shows the AT90S8414 *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 lowermost SRAM addresses, allowing them to be accessed as though they were ordinary memory locations.

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 in-system downloadable Flash memory.

With the relative jump and call instructions, the whole 4K address space is directly accessed. All *AVR* instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-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 SP in the reset routine (before subroutines or interrupts are executed). The 16-bit stack pointer SP is read/write accessible in the I/O space.

The 256 bytes data SRAM can be easily accessed through the four different addressing modes supported in the *AVR* architecture.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters, A/D-converters, and other I/O functions. 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 the different interrupts have a separate interrupt vector in the interrupt vector table at the beginning of the program memory. The different interrupts have priority in accordance with their interrupt vector position. The lower the interrupt address vector the higher priority.

The General Purpose Register File

Figure 5 shows the structure of the 32 general purpose working registers in the CPU.

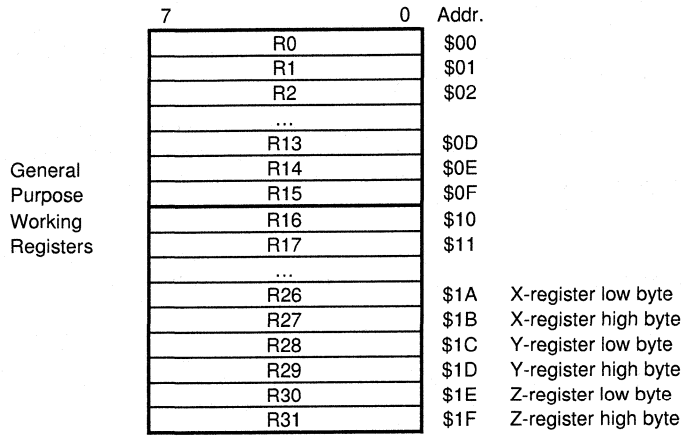


Figure 5: *AVR* CPU general purpose working registers

All the 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 Figure 5, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user SRAM area. 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.

The 256 bytes of SRAM available for general data are implemented as addresses $0020 to $011F.

THE X-REGISTER, Y-REGISTER AND Z-REGISTER

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

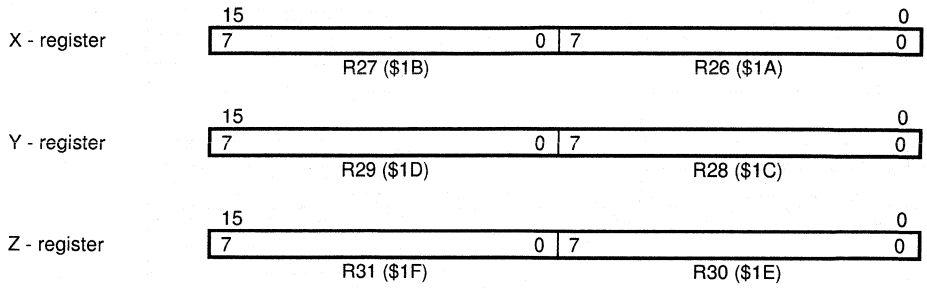


Figure 6: The X, Y and Z Registers

4

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

The ALU - Arithmetic Logic Unit

The high-performance *AVR* ALU operates in direct connection with all the 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, logic and bit-functions. Some microcontrollers in the *AVR* product family feature a hardware multiplier in the arithmetic part of the ALU.

The Downloadable Flash Program Memory

The AT90S8414 contains 8K bytes on-chip downloadable Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 4K x 16 words. The Flash memory has an endurance of at least 1000 write/erase cycles.

See Page 4-69 for a detailed description on Flash data downloading.

Constant tables must be allocated within the address 0-4K (see the LPM - Load Program Memory instruction description).

See Page 4-15 for the different program memory addressing modes.

The SRAM Data Memory - Internal and External

The following figure shows how the AT90S8414 SRAM Memory is organized:

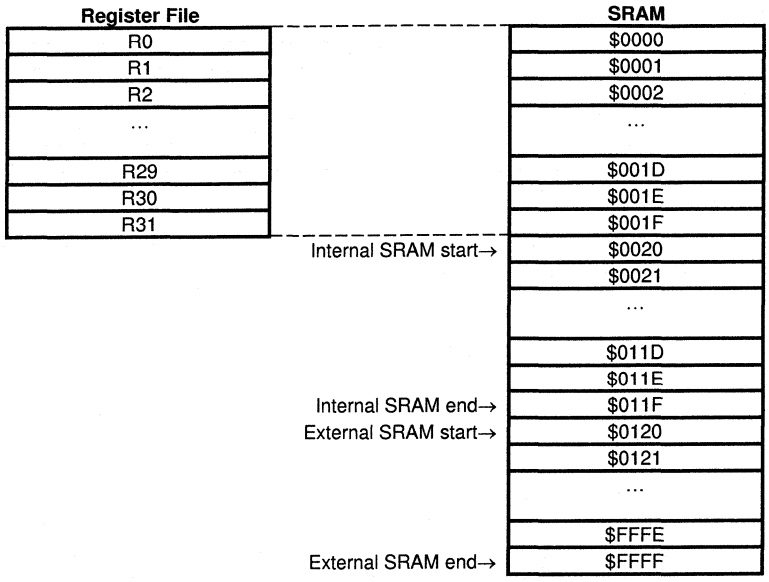


Figure 7: SRAM Organization

The 288 top SRAM Memory locations address both the Register file and the internal data SRAM. The first 32 locations address the register file, and the next 256 locations address the internal data SRAM. An optional external data SRAM can be placed in the same SRAM memory space. Following the last address of the internal SRAM space and up to 64K - 1, is the external data SRAM address area.

When the addresses accessing the SRAM memory space exceeds the internal data SRAM locations, the external data SRAM is accessed using the same instructions as for the internal data SRAM access. When the internal data SRAM is accessed, the read and write strobe pins ($\overline{RD}$ and $\overline{WR}$) are inactive during the whole access cycle. The external data SRAM physical address locations corresponding to the internal data SRAM addresses can not be reached by the CPU. External SRAM operation is enabled by setting the SRE bit in the MCUCR register. See Page 4-28 for details.

The four different addressing modes for the SRAM data memory cover: Direct 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.

The Direct with Displacement mode features a 63 address locations 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, the 256 bytes of internal data SRAM, and the 64K bytes of optional external data SRAM in the AT90S8414 are all accessible through all these addressing modes.

See the next section for a detailed description of the different addressing modes.

The Program and Data Addressing Modes

The AT90S8414 *AVR* Enhanced RISC microcontroller supports powerful and efficient addressing modes for access to the program memory (Flash) and data memory (SRAM). This section describes the different addressing modes supported by the *AVR* architecture. In the figures, OP means the operation code part of the instruction word. To simplify, not all figures show the exact location of the addressing bits.

REGISTER DIRECT, SINGLE REGISTER Rd

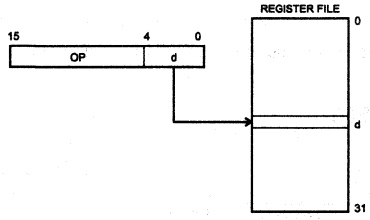


Figure 8: Direct Single Register Addressing

The operand is contained in register d (Rd).

REGISTER DIRECT, TWO REGISTERS Rd AND Rr

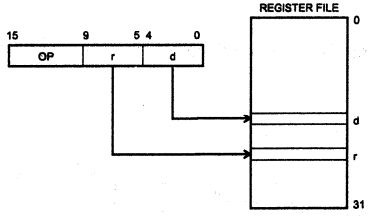


Figure 9: Direct Register Addressing, Two Registers

Operands are contained in register r (Rr) and d (Rd). The result is stored in register d (Rd).

I/O DIRECT

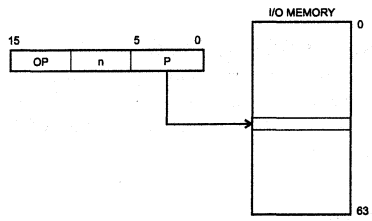


Figure 10: I/O Direct Addressing

Operand address is contained in 6 bits of the instruction word. n is the destination or source register address.

SRAM DIRECT WITH DISPLACEMENT

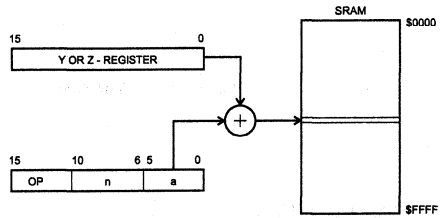


Figure 11: SRAM Direct with Displacement

Operand address is the result of the Y or Z-register contents added to the address contained in 6 bits of the instruction word.

SRAM/REGISTER INDIRECT

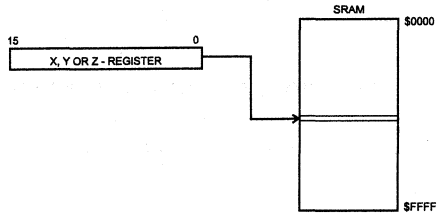


Figure 12: SRAM Indirect Addressing

Operand address is the contents of the X, Y or the Z-register.

SRAM/REGISTER INDIRECT WITH PRE-DECREMENT

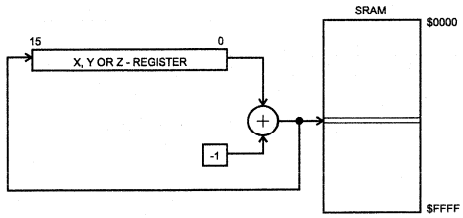


Figure 13: SRAM Indirect Addressing With Pre-Decrement

The X, Y or the Z-register is decremented before the operation. Operand address is the decremented contents of the X, Y or the Z-register.

SRAM/REGISTER INDIRECT WITH POST-INCREMENT

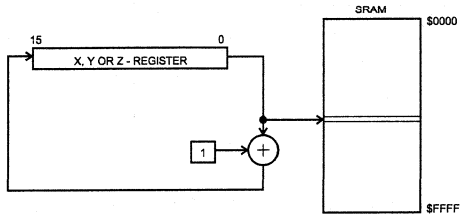


Figure 14: SRAM Indirect Addressing With Post-Increment

The X, Y or the Z-register is incremented after the operation. Operand address is the content of the X, Y or the Z-register prior to incrementing.

CONSTANT ADDRESSING USING THE LPM INSTRUCTION

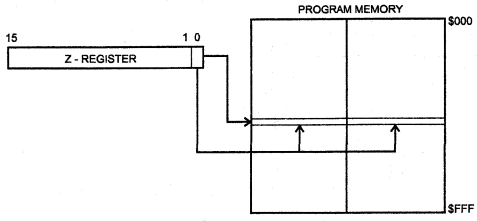


Figure 15: Code Memory Constant Addressing

Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 4K) and LSB, select low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

DIRECT PROGRAM ADDRESS, JMP AND CALL

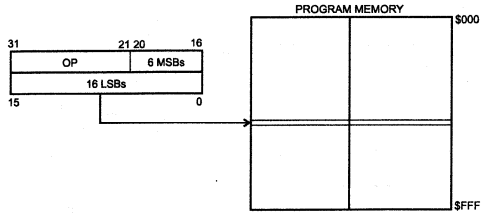


Figure 16: Direct Program Memory Addressing

Program execution continues at the address immediate in the instruction words.

INDIRECT PROGRAM ADDRESSING, IJMP AND ICALL

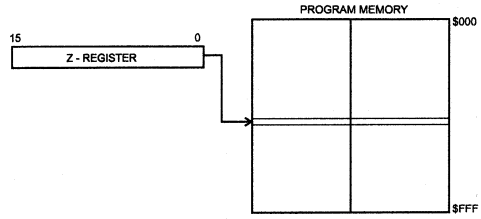


Figure 17: Indirect Program Memory Addressing

Program execution continues at address contained by the Z-register (i.e. the PC is loaded with the contents of the Z-register).

RELATIVE PROGRAM ADDRESSING, RJMP AND RCALL

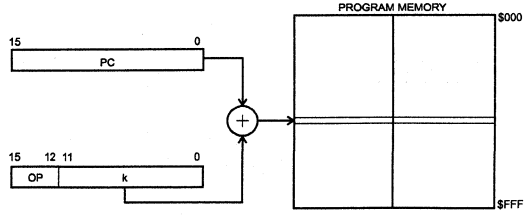


Figure 18: Relative Program Memory Addressing

Program execution continues at address PC + k. The relative address k is ± 2K.

The EEPROM Data Memory

The AT90S8414 contains 256 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on Page 4-40 specifying the EEPROM address register, the EEPROM data register, and the EEPROM control register.

For the SPI data downloading, see Page 4-74 for a detailed description.

Memory Access Times and Instruction Execution Timing

This section describes the general access timing concepts for instruction execution and internal memory access.

The *AVR* CPU is driven by the System Clock Ø, directly generated from the external clock crystal for the chip. No internal clock division is used.

Figure 19 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept to obtain up to 1 MIPS per MHz with the corresponding unique results for functions per cost, functions per clocks, and functions per power-unit.

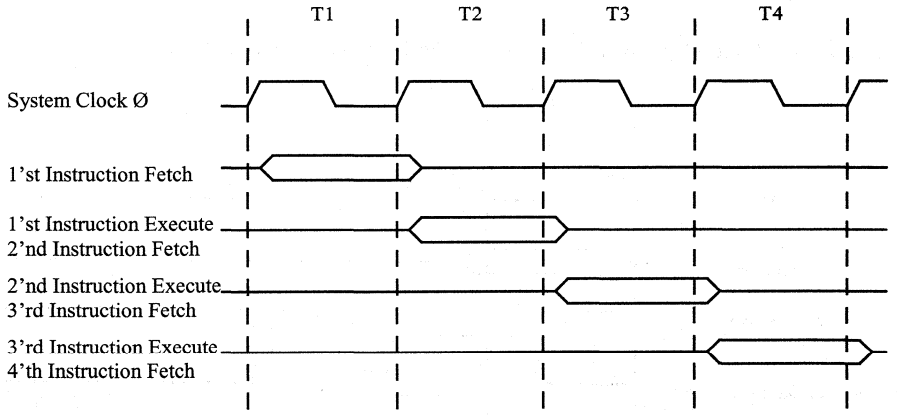


Figure 19: The Parallel Instruction Fetches and Instruction Executions

Figure 20 shows the internal timing concept for the register file. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

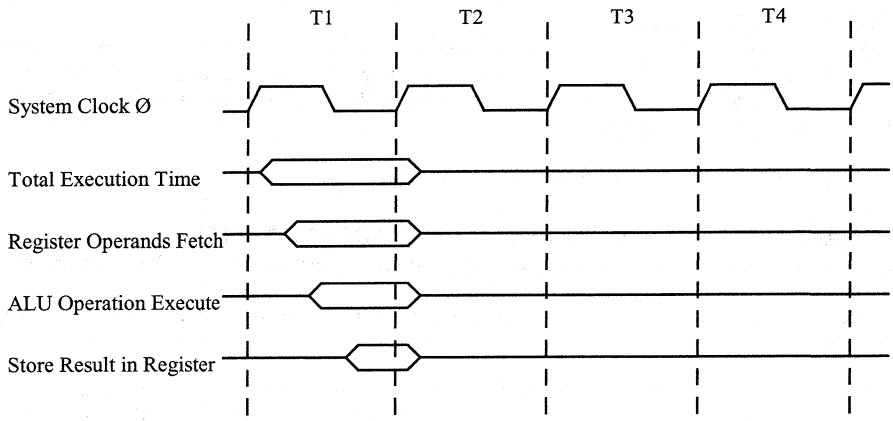


Figure 20: Single Cycle ALU Operation

The internal data SRAM access is performed in two System Clock cycles as described in Figure 21.

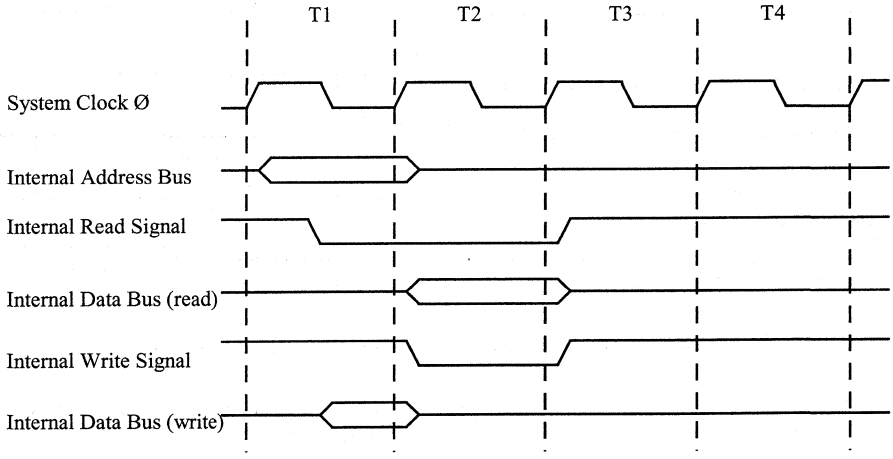


Figure 21: On-Chip Data SRAM Access Cycles

The external data SRAM access is performed in two System Clock cycles as described in Figure 22.

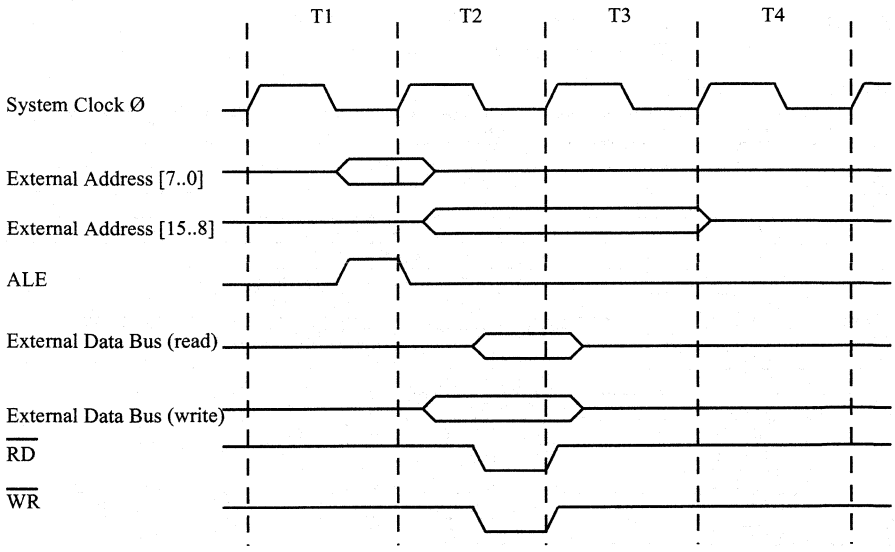


Figure 22: External Data SRAM Memory Cycles without Wait State

The external data SRAM memory access cycle with the Wait State bit enabled (Wait State active) is shown in Figure 23.

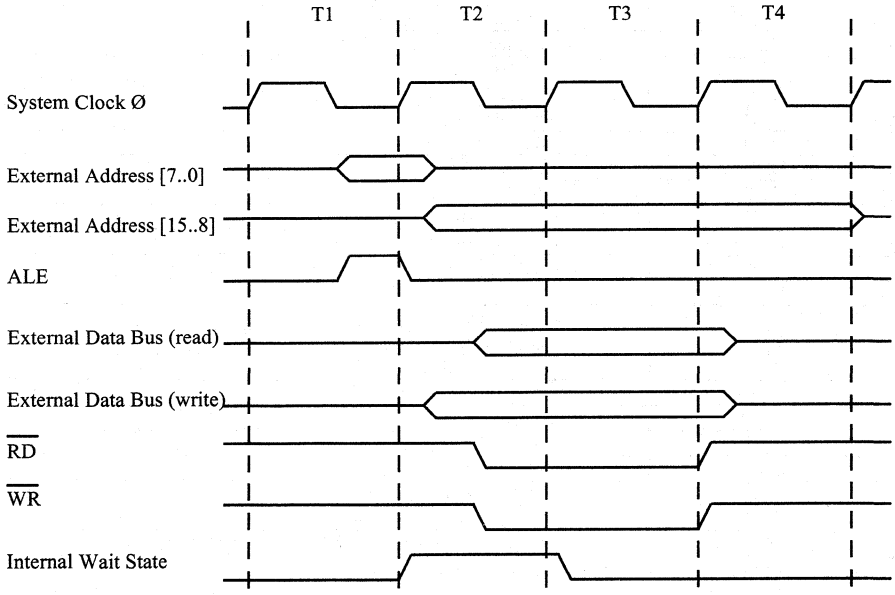


Figure 23: External Data SRAM Memory Cycles with Wait State

I/O Memory

The I/O space definition of the AT90S8414 is shown in the following table:

Table 1: AT90S8414 I/O Space

Address Hex	Name	Function
$3F	SREG	Status REGister
$3E	SPH	Stack Pointer High
$3D	SPL	Stack Pointer Low
$3B	GIMSK	General Interrupt MaSK register
$39	TIMSK	Timer/Counter Interrupt MaSK register
$38	TIFR	Timer/Counter Interrupt Flag register
$35	MCUCR	MCU general Control Register
$33	TCCR0	Timer/Counter0 Control Register
$32	TCNT0	Timer/Counter0 (8-bit)
$31	OCR0	Timer/Counter0 Output Compare Register
$2F	TCCR1A	Timer/Counter1 Control Register A
$2E	TCCR1B	Timer/Counter1 Control Register B
$2D	TCNT1H	Timer/Counter1 High Byte
$2C	TCNT1L	Timer/Counter1 Low Byte
$2B	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
$2A	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
$29	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
$28	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
$25	ICR1H	T/C 1 Input Capture Register High Byte
$24	ICR1L	T/C 1 Input Capture Register Low Byte
$21	WDTCR	Watchdog Timer Control Register
$1E	EEAR	EEPROM Address Register
$1D	EEDR	EEPROM Data Register
$1C	EECR	EEPROM Control Register
$1B	PORTA	Data Register, Port A
$1A	DDRA	Data Direction Register, Port A
$19	PINA	Input Pins, Port A
$18	PORTB	Data Register, Port B
$17	DDRB	Data Direction Register, Port B
$16	PINB	Input Pins, Port B
$15	PORTC	Data Register, Port C
$14	DDRC	Data Direction Register, Port C
$13	PINC	Input Pins, Port C
$12	PORTD	Data Register, Port D
$11	DDRD	Data Direction Register, Port D
$10	PIND	Input Pins, Port D
$0F	SPDR	SPI I/O Data Register
$0E	SPSR	SPI Status Register
$0D	SPCR	SPI Control Register
$0C	UDR	UART I/O Data Register
$0B	USR	UART Status Register
$0A	UCR	UART Control Register
$09	UBRR	UART Baud Rate Register
$08	ACSR	Analog Comparator Control and Status Register

Note: reserved and unused locations are not shown in the table

All the different AT90S8414 I/Os and peripherals are placed in the I/O space. The different 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.

The different I/O and peripherals control registers are explained in the following chapters.

THE STATUS REGISTER - SREG

The *AVR* status register - SREG - at I/O space location $3F is defined as:

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

Bit 7 - I : Global Interrupt Enable:
The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in the interrupt mask registers - GIMSK and TIMSK. If the global interrupt enable register is cleared (zero), none of the interrupts are enabled independent of the GIMSK and TIMSK values. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

Bit 6 - T : Bit Copy Storage:
The bit copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T bit as source and destination for the operated bit. A bit from a register in the register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the register file by the BLD instruction.

Bit 5 - H : Half Carry Flag:
The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

Bit 4 - S : Sign Bit, $S = N \oplus V$:
The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

Bit 3 - V : Two's Complement Overflow Flag:
The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

Bit 2 - N : Negative Flag:
The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 1 - Z : Zero Flag:
The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

Bit 0 - C : Carry Flag:
The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information.

THE STACK POINTER - SP

The general *AVR* 16-bit Stack Pointer is effectively built up of two 8-bit registers in the I/O space locations $3E and $3D. Since the stack address space is within the data SRAM, a 9 bit stack pointer is used to address the stack in the AT90S8414.

Bit	15	14	13	12	11	10	9	8	
$3E	-	-	-	-	-	-	-	SP8	SPH
$3D	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The Stack Pointer points to the data SRAM stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when data is pushed onto the Stack with subroutine CALL and interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt IRET.

Reset and Interrupt Handling

The AT90S8414 provides 13 different interrupt sources. These interrupts and the separate reset vector, each have a separate program vector in the program memory space. 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 2. 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 External Interrupt Request 0 etc.

Table 2: Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	Hardware Pin and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	INT1	External Interrupt Request 1
4	$003	TIMER1 CAPT	Timer/Counter1 Capture Event
5	$004	TIMER1 COMPA	Timer/Counter1 Compare Match A
6	$005	TIMER1 COMPB	Timer/Counter1 Compare Match B
7	$006	TIMER1 OVF	Timer/Counter1 Overflow
8	$007	TIMER0, COMP	Timer/Counter0 Compare Match
9	$008	TIMER0, OVF	Timer/Counter0 Overflow
10	$009	SPI, STC	Serial Transfer Complete
11	$00A	UART, RX	UART, Rx Complete
12	$00B	UART, UDRE	UART Data Register Empty
13	$00C	UART, TX	UART, Tx Complete
14	$00D	ANA_COMP	Analog Comparator

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

```
Address     Labels      Code                Comments
$000                    rjmp  RESET         ; Reset Handle
$001                    rjmp  EXT_INT0      ; IRQ0 Handle
$002                    rjmp  EXT_INT1      ; IRQ1 Handle
$003                    rjmp  TIM1_CAPT     ; Timer1 capture Handle
$004                    rjmp  TIM1_COMPA    ; Timer1 compareA Handle
$005                    rjmp  TIM1_COMPB    ; Timer1 compareB Handle
$006                    rjmp  TIM1_OVF      ; Timer1 overflow Handle
$007                    rjmp  TIM0_COMP     ; Timer0 compare Handle
$008                    rjmp  TIM0_OVF      ; Timer0 overflow Handle
$009                    rjmp  SPI_HANDLE    ; SPI TX Handle
$00a                    rjmp  UART_RXC      ; UART RX Complete Handle
$00b                    rjmp  UART_DRE      ; UDR Empty Handle
$00c                    rjmp  UART_TXC      ; UART TX Complete Handle
$00d                    rjmp  ANA_COMP      ; Analog Comparator Handle
;
$00e        MAIN:       <instr>  xxx        ; Main program start
  ...         ...         ...      ...
```

RESET

A Reset State is enabled by a high level on the RESET pin. The pin must be held high for at least two crystal clock cycles. All internal 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 RJMP - relative jump - 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.

4

INTERRUPT HANDLING

The AT90S8414 has two 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 must set (one) the I-bit to enable interrupts.

When the Program Counter is vectored to the actual interrupt vector in order to execute the interrupt handling routine, hardware 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.

THE GENERAL INTERRUPT MASK REGISTER - GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B	INT1	INT0	-	-	-	-	-	-	GIMSK
Read/Write	R/W	R/W	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - INT1 : External Interrupt Request 1 Enable:

When the INT1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is enabled. The Interrupt Sense Control bits 1/0 (ISC11 and ISC10) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT1 pin or level sensed. If the INT1 pin for external interrupts shall be activated, the DDD3 bit in the Data Direction Register PORTD (DDRD) must be cleared (zero) to force an input pin. The corresponding interrupt of External Interrupt Request 1 is executed from program memory address $002. See also "External Interrupts".

Bit 6 - INT0 : External Interrupt Request 0 Enable:
When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) defines whether the external interrupt is activated on rising or falling edge of the INT0 pin or level sensed. If the INT0 pin for external interrupts shall be activated, the DDD2 bit in the Data Direction Register PORTD (DDRD) must be cleared (zero) to force an input pin. The corresponding interrupt of External Interrupt Request 0 is executed from program memory address $001. See also "External Interrupts".

Bits 5..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and always read as zero.

THE TIMER/COUNTER INTERRUPT MASK REGISTER - TIMSK

Bit	7	6	5	4	3	2	1	0	
$39	TOIE1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE0	OCIE0	TIMSK
Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - TOIE1 : Timer/Counter1 Overflow Interrupt Enable:
When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $006) is executed if an overflow in Timer/Counter1 occurs. The Overflow Flag (Timer/Counter1) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR. When Timer/Counter1 is in PWM mode, the Timer Overflow flag is set when the counter changes counting direction at $0000.

Bit 6 - OCE1A :Timer/Counter1 Output CompareA Match Interrupt Enable:
When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareA Match interrupt is enabled. The corresponding interrupt (at vector $004) is executed if a CompareA match in Timer/Counter1 occurs. The CompareA Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 5 - OCIE1B :Timer/Counter1 Output CompareB Match Interrupt Enable:
When the OCIE1B bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 CompareB Match interrupt is enabled. The corresponding interrupt (at vector $005) is executed if a CompareB match in Timer/Counter1 occurs. The CompareB Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 4 - Res : Reserved bit:
This bit is a reserved bit in the AT90S8414 and always reads zero.

Bit 3 - TICIE1 : Timer/Counter1 Input Capture Interrupt Enable:
When the TICIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Input Capture Event Interrupt is enabled. The corresponding interrupt (at vector $003) is executed if a capture-triggering event occurs on pin 31, ICP. The Input Capture Flag in Timer/Counter1 is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 2 - Res : Reserved bit:
This bit is a reserved bit in the AT90S8414 and always reads zero.

Bit 1 - TOIE0 : Timer/Counter0 Overflow Interrupt Enable:
When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $008) is executed if an overflow in Timer/Counter0 occurs. The Overflow Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

Bit 0 - OCIE0 :Timer/Counter0 Output Compare Match Interrupt Enable:
When the OCIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Compare Match interrupt is enabled. The corresponding interrupt (at vector $007) is executed if a compare match in Timer/Counter0 occurs. The Compare Flag (Timer0) is set (one) in the Timer/Counter Interrupt Flag Register - TIFR.

THE TIMER/COUNTER INTERRUPT FLAG REGISTER - TIFR

Bit	7	6	5	4	3	2	1	0	
$38	TOV1	OCF1A	OCIFB	-	ICF1	-	TOV0	OCF0	TIFR
Read/Write	R/W	R/W	R/W	R	R/W	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - TOV1 : Timer/Counter1 Overflow Flag:
The TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logic one to the flag. When the I-bit in SREG, and TOIE1 (Timer/Counter1 Overflow Interrupt Enable), and TOV1 are set (one), the Timer/Counter1 Overflow Interrupt is executed. In PWM mode, this bit is set when Timer/Counter1 changes counting direction at $0000.

Bit 6 - OCF1A : Output Compare Flag 1A:
The OCF1A bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1A - Output Compare Register 1A. OCF1A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1A is cleared by writing a logic one to the flag. When the I-bit in SREG, and OCIE1A (Timer/Counter1 Compare match InterruptA Enable), and the OCF1A are set (one), the Timer/Counter1 Compare match Interrupt is executed.

Bit 5 - OCF1B : Output Compare Flag 1B:
The OCF1B bit is set (one) when compare match occurs between the Timer/Counter1 and the data in OCR1B - Output Compare Register 1B. OCF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF1 is cleared by writing a logic one to the flag.. When the I-bit in SREG, and OCIE1B (Timer/Counter1 Compare match InterruptB Enable), and the OCF1B are set (one), the Timer/Counter1 Compare match Interrupt is executed.

Bit 4 - Res : Reserved bit:
This bit is a reserved bit in the AT90S8414 and always reads zero.

Bit 3 - ICF1 : - Input Capture Flag 1:
The ICF1 bit is set (one) to flag an input capture event, indicating that the Timer/Counter1 value has been transferred to the input capture register - ICR1. ICF1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ICF1 is cleared by writing a logic one to the flag.

Bit 2 - Res : Reserved bit:
This bit is a reserved bit in the AT90S8414 and always reads zero.

Bit 1 - TOV0 : Timer/Counter0 Overflow Flag:
The bit TOV0 is set (one) when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, and TOIE0 (Timer/Counter0 Overflow Interrupt Enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed.

Bit 0 - OCF0 : Output Compare Flag 0:
The OCF0 bit is set (one) when a compare match occurs between the Timer/Counter0 and the compared data in OCR0 - Output Compare Register0. OCF0 is cleared by hardware when executing the corresponding interrupt handling vector.

Alternatively, OCF0 is cleared by writing a logic one to the flag.. When the SREG I-bit, and OCIE0 (Timer/Counter0 Compare match Interrupt Enable), and OCF0 are set (one), the Timer/Counter0 Compare match Interrupt is executed.

EXTERNAL INTERRUPTS

The external interrupts are triggered by the INT1 and INT0 pins. Since these pins are alternate function pins in the general I/O ports, the corresponding pins must be set as input pins in the data direction register - DDRX.

The external interrupts are set up as indicated in the specification for the general interrupt mask register - GIMSK.

INTERRUPT RESPONSE TIME

The interrupt response time for all the enabled *AVR* interrupt is 4 clock cycles. After the 4 clock cycles the program vector address for the actual interrupt handling routine is executed. During this 4 clock cycle period, the Program Counter (2 bytes) is pushed onto the Stack, and the Stack Pointer is decremented by 2. A return from an interrupt handling routine (same as for a subroutine call routine) takes 4 clock cycles. During these 4 clock cycles, the Program Counter (2 bytes) is popped back from the Stack, and the Stack Pointer is incremented by 2.

Note that the Status Register - SREG - is not handled by the *AVR* hardware, neither for interrupts nor for subroutines. For the interrupt handling routines requiring a storage of the SREG, this must be performed by user software.

For Interrupts triggered by events that can remain static (E.g. the Output Compare register0 matching the value of Timer/Counter0) 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.

MCU CONTROL REGISTER - MCUCR

The MCU Control Register contains control bits for general MCU functions.

Bit	7	6	5	4	3	2	1	0	
$35	SRE	SRW	SE	SM	ISC11	ISC10	ISC01	ISC00	MCUCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - SRE : External SRAM Enable:
When the SRE bit is set (one), the external data SRAM is enabled, and the pin functions AD0-7 (Port A), A8-15 (Port C), $\overline{WR}$ and $\overline{RD}$ (Port D) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective data direction registers. See "The SRAM Data Memory - Internal and External" for description of the External SRAM pin functions. When the SRE bit is cleared (zero), the external data SRAM is disabled, and the normal pin and data direction settings are used.

Bit 6 - SRW : External SRAM Wait State:
When the SRW bit is set (one), a one cycle wait state is inserted in the external data SRAM access cycle. When the SRW bit is cleared (zero), the external data SRAM access is executed with the normal two cycle scheme. See Figure 22 External Data SRAM Memory Cycles without Wait State and Figure 23: External Data SRAM Memory Cycles with Wait State.

Bit 5 - SE : Sleep Enable:
The SE bit must be set (one) to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to set the Sleep Enable SE bit just before the execution of the SLEEP instruction.

Bit 4 - SM : Sleep Mode:
This bit selects between the two available sleep modes

When cleared (zero) the SM control bit forces the MCU into the Idle Mode stopping the CPU but allowing the General Purpose Working Registers, SRAM, Timer/Counters, SPI port, and interrupt system to continue operating.

When set (one), the SM control bit forces the MCU into the Power Down Mode saving the General Purpose Working Registers and the SRAM data, but freezing the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

To enter the sleep modes, the SE bit must be set (one) and a SLEEP instruction must be executed. The instruction following SLEEP is executed before entering sleep mode. If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU awakes, executes the interrupt routine and resumes execution from the address following the last executed instruction. If reset occurs while the MCU is in Sleep Mode, the MCU awakes and executes from the reset vector.

Bit 3, 2 - ISC11, ISC10 : Interrupt Sense Control 1 bit 1 and bit 0:
The External Interrupt 1 is activated by the external pin INT1 if the SREG I-flag and the corresponding interrupt mask in the GIMSK is set. The level and edges on the external INT1 pin that activate the interrupt are defined in the following table:

Table 3: Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Note: When changing the ISC11/ISC10 bits, INT1 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bit 1, 0 - ISC01, ISC00 : Interrupt Sense Control 0 bit 1 and bit 0:
The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask is set. The level and edges on the external INT0 pin that activate the interrupt are defined in the following table:

Table 4: Interrupt 0 Sense Control

ISC01	ISC00	Description
0	0	The low level of INT0 generates an interrupt request.
0	1	Reserved
1	0	The falling edge of INT0 generates an interrupt request.
1	1	The rising edge of INT0 generates an interrupt request.

Note: When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Timer / Counters

The AT90S8414 provides two general purpose Timer/Counters - one 8-bit T/C and one 16-bit T/C. The Timer/Counters have separate prescaling selection from the same 10-bit prescaling timer. Both Timer/Counters can either be used as a timer with an internal clock timebase or as a counter with an external pin connection which triggers the counting.

The Timer/Counter Prescaler

Figure 24 shows the general Timer/Counter prescaler.

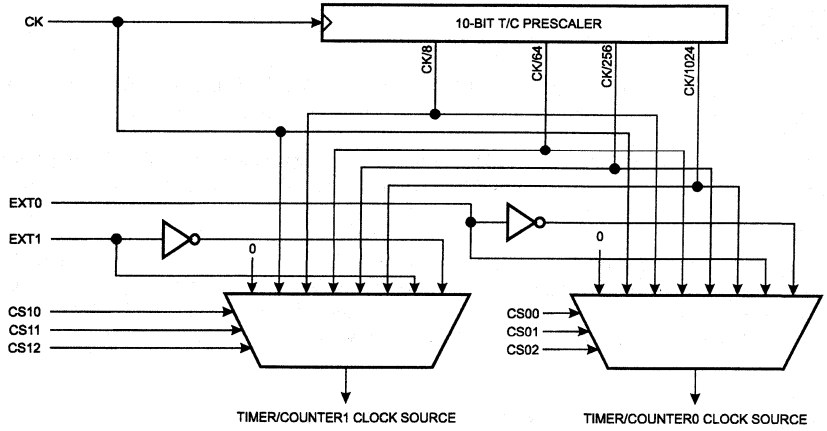


Figure 24: Timer/Counter Prescaler

The four different prescaled selections are: CK/8, CK/64, CK/256 and CK/1024 where CK is the oscillator clock. For the two Timer/Counters, added selections as CK, external source and stop, can be selected as clock sources.

The 8-Bit Timer/Counter0

Figure 25 shows the block diagram for Timer/Counter0.

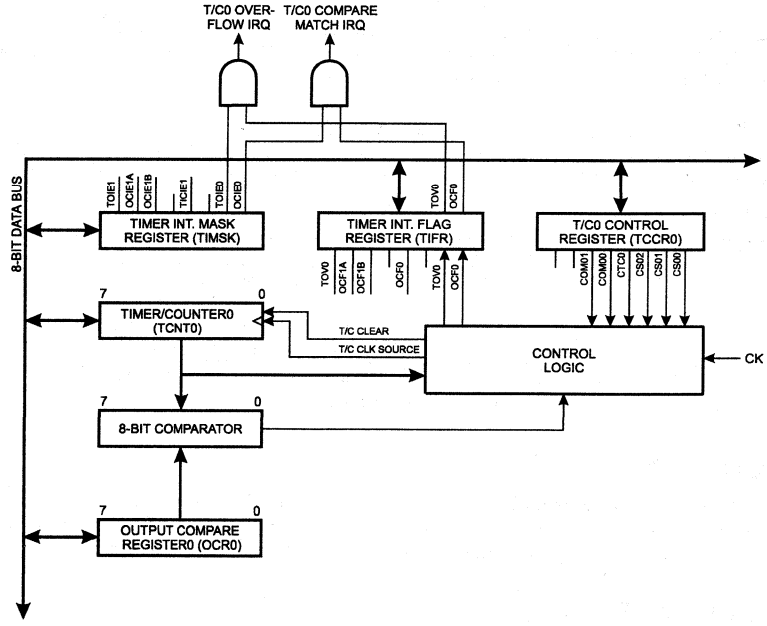


Figure 25: Timer/Counter0 Block Diagram

The 8-bit Timer/Counter0 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter0 Control Register - TCCR0. The different status flags (overflow and compare match) and control signals are found in the Timer/Counter0 Control Register - TCCR0. The interrupt enable/disable settings for Timer/Counter0 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter0 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time for the external clock being low and high must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 8-bit Timer/Counter0 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities make the Timer/Counter0 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter0 supports an Output Compare function using the Output Compare Register 0 - OCR0 as the data source to be compared to the Timer/Counter0 contents. The Output Compare functions include optional clearing of the counter on compare matches, and actions on the Output Compare pin 0 on compare matches. The Output Compare pin 0 function makes the Timer/Counter0 useful for PWM (Pulse Width Modulation) functions.

THE TIMER/COUNTER0 CONTROL REGISTER - TCCR0

Bit	7	6	5	4	3	2	1	0	
$33	-	-	COM01	COM00	CTC0	CS02	CS01	CS00	TCCR0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7,6 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and always read zero.

Bits 5,4 - COM01, COM00 : Compare Output Mode0, bit 1 and 0:
The COM01 and COM00 control bits determine any output pin action following a compare match in Timer/Counter0. Any output pin actions are effective on pin OC0 - Output Compare pin 0. Since this is an alternative function to an I/O port, the corresponding data direction control bit must be set (one) to control an output pin. The control configuration is defined in the following table:

Table 5: Compare 0 Mode Select

COM01	COM00	Description
0	0	Timer/Counter0 disconnected from output pin OC0.
0	1	Toggle the OC0 output line.
1	0	Clear the OC0 output line (to zero).
1	1	Set the OC0 output line (to one).

Note: When changing the COM01/COM00 bits, Output Compare Interrupt 0 must be disabled by clearing its Interrupt Enable bit in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bit 3 - CTC0 : Clear Timer/Counter0 on Compare match:
When the CTC0 control bit is set (one), the Timer/Counter0 is reset to $00 in the clock cycle after the compare match. If the CTC0 control bit is cleared, the Timer/Counter0 continues running freely until it is stopped, set, cleared or wraps around (overflow).

Bits 2,1,0 - CS02, CS01, CS00 : Clock Select0, bit 2,1 and 0:
The Clock Select0 bits 2,1 and 0 define the prescaling source of Timer0.

Table 6: Clock 0 Prescale Select

CS02	CS01	CS00	Description
0	0	0	Stop, the Timer/Counter0 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T0, rising edge
1	1	1	External Pin T0, falling edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual data direction control register (cleared to zero gives an input pin).

Bits 5..3 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and always read zero.

THE TIMER COUNTER 0 - TCNT0

Bit	7	6	5	4	3	2	1	0	
$32	MSB							LSB	TCNT0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Timer/Counter0 is realized as an up-counter with read and write access. If the Timer/Counter0 is written and a clock source is present, the Timer/Counter0 continues counting in the clock cycle following the write operation.

THE OUTPUT COMPARE REGISTER 0 - OCR0

Bit	7	6	5	4	3	2	1	0	
$31	MSB							LSB	OCR0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The Output Compare Register 0 is the source register for the Timer/Counter0 compare match functions.

The 16-Bit Timer/Counter1

Figure 26 shows the block diagram for Timer/Counter1.

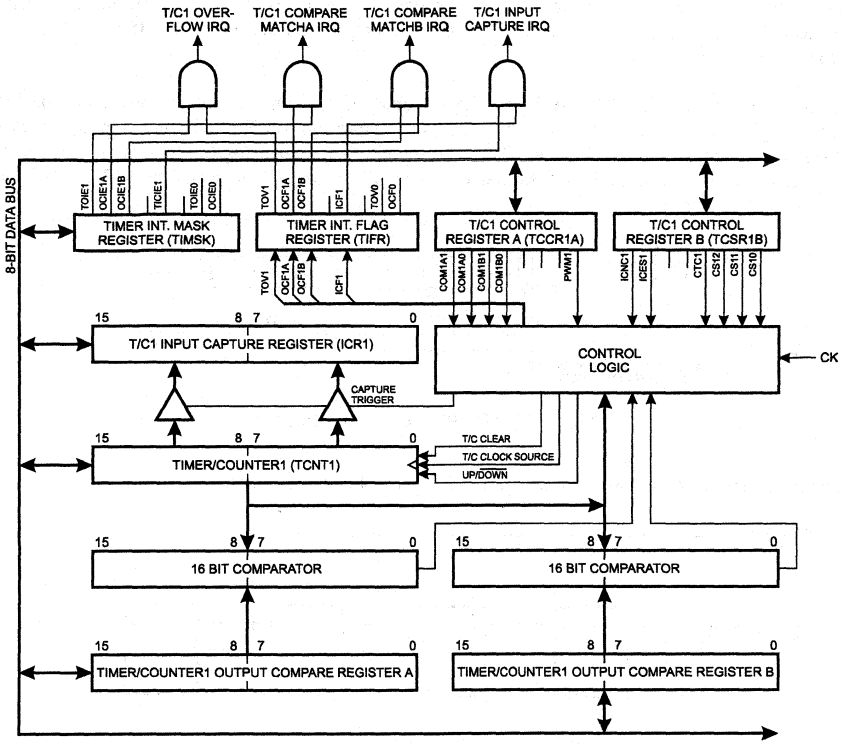


Figure 26: Timer/Counter1 Block Diagram

The 16-bit Timer/Counter1 can select clock source from CK, prescaled CK, or an external pin. In addition it can be stopped as described in the specification for the Timer/Counter1 Control Register - TCCR1B. The different status flags (overflow, compare match and capture event) and control signals are found in the Timer/Counter1 Control Registers - TCCR1A and TCCR1B. The interrupt enable/disable settings for Timer/Counter1 are found in the Timer/Counter Interrupt Mask Register - TIMSK.

When Timer/Counter1 is externally clocked, the external signal is synchronized with the oscillator frequency of the CPU. To assure proper sampling of the external clock, the minimum time for the external clock being low and high must be at least one internal CPU clock period. The external clock signal is sampled on the rising edge of the internal CPU clock.

The 16-bit Timer/Counter1 features both a high resolution and a high accuracy usage with the lower prescaling opportunities. Similarly, the high prescaling opportunities makes the Timer/Counter1 useful for lower speed functions or exact timing functions with infrequent actions.

The Timer/Counter1 supports two Output Compare functions using the Output Compare Register 1 A and B - OCR1A and OCR1B as the data sources to be compared to the Timer/Counter1 contents. The Output Compare functions include optional clearing of the counter on compareA match, and actions on the Output Compare pins on both compare matches.

Timer/Counter1 can also be used as a 10 bit Pulse With Modulator. In this mode the counter and the OCR1A/OCR1B registers serve as a dual glitch-free stand-alone PWM with centered pulses. Refer to Page 4-38 for a detailed description on this function.

The Input Capture function of Timer/Counter1 provides a capture of the Timer/Counter1 contents to the Input Capture Register - ICR1, triggered by an external event on the Input Capture Pin - ICP. The actual capture event settings are defined by the Timer/Counter1 Control Register - TCCR1B. In addition, the Analog Comparator can be set to trigger the Input Capture. Refer to the paragraph, "The Analog Comparator", for details on this. The ICP pin logic is shown in Figure 27.

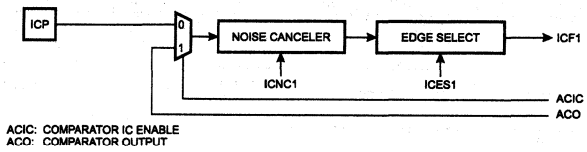


Figure 27: ICP Pin Schematic Diagram

The Timer/Counter1 input capture noise canceler block diagram is shown in Figure 28.

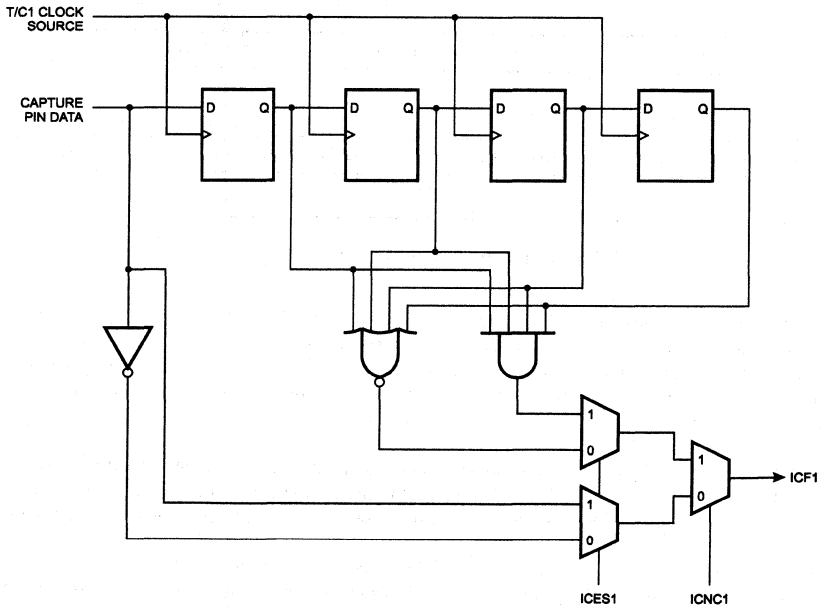


Figure 28: The Input Capture Noise Canceler

If the noise canceler function is enabled, the actual trigger condition for the capture event is monitored over 4 samples before the capture is activated. The sampling clock is the same clock as the clock source selected for the Timer/Counter1.

THE TIMER/COUNTER1 CONTROL REGISTER A - TCCR1A

Bit	7	6	5	4	3	2	1	0	
$2F	COM1A1	COM1A0	COM1B1	COM1B0	-	-	-	PWM1	TCCR1A
Read/Write	R/W	R/W	R/W	R/W	R	R	R	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7,6 - COM1A1, COM1A0 : Compare Output Mode1A, bits 1 and 0:
The COM1A1 and COM1A0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1A - Output CompareA pin 1. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The control configuration is shown in Table 7.

Bits 5,4 - COM1B1, COM1B0 : Compare Output Mode1B, bits 1 and 0:
The COM1B1 and COM1B0 control bits determine any output pin action following a compare match in Timer/Counter1. Any output pin actions affect pin OC1B - Output CompareB. Since this is an alternative function to an I/O port, the corresponding direction control bit must be set (one) to control an output pin. The following control configuration is given:

Table 7: Compare 1 Mode Select

COM1X1	COM1X0	Description
0	0	Timer/Counter1 disconnected from output pin OC1X
0	1	Toggle the OC1X output line.
1	0	Clear the OC1X output line (to zero).
1	1	Set the OC1X output line (to one).

Notes: X = A or B
In PWM mode, these bits have a different function. Refer to Table 9 for a detailed description.
When changing the COM1X1/COM1X0 bits, Output Compare Interrupts 1 must be disabled by clearing their Interrupt Enable bits in the TIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Bits 3..1 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and always read zero.

Bit 0 - PWM1 : Pulse Width Modulator enable:
This bit enables the PWM mode for Timer/Counter1. This mode is described on Page 4-38.

THE TIMER/COUNTER1 CONTROL REGISTER B - TCCR1B

Bit	7	6	5	4	3	2	1	0	
$2E	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	TCSR1B
Read/Write	R/W	R/W	R	R	R/w	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - ICNC1 : Input Capture1 Noise Canceler (4 CKs):
When the ICNC1 bit is cleared (zero), the input capture trigger noise canceler function is disabled. The input capture is triggered at the first rising/falling edge sampled on the ICP - input capture pin - as specified. When the ICNC1 bit is set (one), four successive samples are measures on the ICP - input capture pin, and all samples must be high/low according to the input capture trigger specification in the ICES1 bit. The actual sampling frequency is the same as the clock source frequency selected for the Timer/Counter1.

Bit 6 - ICES1 : Input Capture1 Edge Select:
While the ICES1 bit is cleared (zero), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the falling edge of the input capture pin - ICP. While the ICES1 bit is set (one), the Timer/Counter1 contents are transferred to the Input Capture Register - ICR1 - on the rising edge of the input capture pin - ICP.

Bits 5, 4 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and always read zero.

Bit 3 - CTC1 : Clear Timer/Counter1 on Compare match:
When the CTC1 control bit is set (one), the Timer/Counter1 is reset to $0000 in the clock cycle after a compareA match. If the CTC1 control bit is cleared, the Timer/Counter1 continues counting until it is stopped, cleared, wraps around (overflow) or changes direction. In PWM mode, this bit has no effect.

Bits 2,1,0 - CS12, CS11, CS10 : Clock Select1, bit 2,1 and 0:
The Clock Select1 bits 2,1 and 0 define the prescaling source of Timer/Counter1.

Table 8: Clock 1 Prescale Select

CS12	CS11	CS10	Description
0	0	0	Stop, the Timer/Counter1 is stopped.
0	0	1	CK
0	1	0	CK / 8
0	1	1	CK / 64
1	0	0	CK / 256
1	0	1	CK / 1024
1	1	0	External Pin T1, rising edge
1	1	1	External Pin T1, falling edge

The Stop condition provides a Timer Enable/Disable function. The CK down divided modes are scaled directly from the CK oscillator clock. If the external pin modes are used, the corresponding setup must be performed in the actual direction control register (cleared to zero gives an input pin).

THE TIMER/COUNTER1 - TCNT1H AND TCNT1L

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

This 16-bit register contains the prescaled value of the 16-bit Timer/Counter1. To ensure that both the high and low bytes are read and written simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary register (TEMP).

- **TCNT1 Timer/Counter1 Write:**
  When the CPU writes to the low byte TCNT1L, the written data is placed in the TEMP register. Next, when the CPU writes the high byte TCNT1H, this byte of data is combined with the byte data in the TEMP register, and all 16 bits are written in the TCNT1 Timer/Counter1 register simultaneously. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register write operation.

- **TCNT1 Timer/Counter1 Read:**
  When the CPU reads the low byte TCNT1L, the data of the low byte TCNT1L is sent to the CPU and the data of the high byte TCNT1H is placed in the TEMP register. When the CPU reads the data in the high byte TCNT1H, the CPU receives the data in the TEMP register. Consequently, the low byte TCNT1L must be accessed first for a full 16-bit register read operation.

The Timer/Counter1 is realized as an up or up/down (in PWM mode) counter with read and write access. If Timer/Counter1 is written to and a clock source is selected, the Timer/Counter1 continues counting in the clock cycle after it is preset with the written value.

TIMER/COUNTER1 OUTPUT COMPARE REGISTER - OCR1AH AND OCR1AL

Bit	15	14	13	12	11	10	9	8	
$2B	MSB								OCR1AH
$2A								LSB	OCR1AL
	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

TIMER/COUNTER1 OUTPUT COMPARE REGISTER - OCR1BH AND OCR1BL

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

The output compare registers are 16-bit read/write registers.

The Timer/Counter1 Output Compare Registers contain the data to be continuously compared with Timer/Counter1. Actions on compare matches are specified in the Timer/Counter1 Control and Status register.

Since the Output Compare Registers - OCR1A and OCR1B - are 16-bit registers, a temporary register TEMP is used when OCR1A/B are written to ensure that both bytes are updated simultaneously. When the CPU writes the low byte, OCR1AL or OCR1BL, the data is temporarily stored in the TEMP register. When the CPU writes the high byte, OCR1AH or OCR1BH, the TEMP register is simultaneously written to OCR1AL or OCR1BL. Consequently, the low byte OCR1AL or OCR1BL must be written first for a full 16-bit register write operation.

THE TIMER/COUNTER1 INPUT CAPTURE REGISTER - ICR1H AND ICR1L

Bit	15	14	13	12	11	10	9	8	
$25	MSB								ICR1H
$24								LSB	ICR1L
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R	R	R	R	
	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

The input capture register is a 16-bit read-only register.

When the rising or falling edge (according to the input capture edge setting - ICES1) of the signal at the input capture pin - ICP - is detected, the current value of the Timer/Counter1 is transferred to the Input Capture Register - ICR1. At the same time, the input capture flag - ICF1 - is set (one).

Since the Input Capture Register - ICR1 - is a 16-bit register, a temporary register TEMP is used when ICR1 is read to ensure that both bytes are read simultaneously. When the CPU reads the low byte ICR1L, the data is sent to the CPU and the data of the high byte ICR1H is placed in the TEMP register. When the CPU reads the data in the high byte ICR1H, the CPU receives the data in the TEMP register. Consequently, the low byte ICR1L must be accessed first for a full 16-bit register read operation.

TIMER/COUNTER1 IN PWM MODE

When the PWM mode is selected, Timer/Counter1 and the Output Compare Register1A - OCR1A and the Output Compare Register1B - OCR1B, form a dual 10-bit, free-running, glitch-free and phase correct PWM with outputs on the PD5(OC1A) and OC1B pins. Timer/Counter1 acts as an up/down counter, counting up from $0000 to $03FF (i.e. from 0 to 1023), when it turns and counts down again to zero before the cycle is repeated. When the counter value matches the contents of the 10 least significant bits of OCR1A or OCR1B, the PD5(OC1A)/OC1B pins are set or cleared according to the settings of the COM1A1/COM1A0 or COM1B1/COM1B0 bits in the Timer/Counter1 Control Register TCCR1A. Refer to Table 9 for details.

Table 9: Compare1 Mode Select in PWM Mode

COM1X1	COM1X0	Effect on OCX1
0	0	Not connected
0	1	Not connected
1	0	Cleared on compare match, upcounting. Set on compare match, downcounting (non-inverted PWM).
1	1	Cleared on compare match, downcounting. Set on compare match, upcounting (inverted PWM).

Note: X = A or B

Note that in the PWM mode, the 10 least significant OCR1A/OCR1B bits, when written, are transferred to a temporary location. They are latched when Timer/Counter1 reaches the top - $03FF. This prevents the occurrence of odd-length PWM pulses (glitches) in the event of an unsynchronized OCR1A/OCR1B write. See Figure 29 for an example.

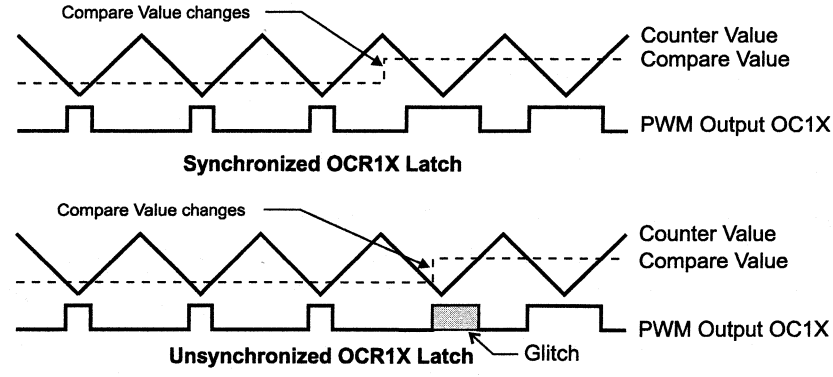


Figure 29: Effects on Unsynchronized OCR1 Latching

When OCR1 contains $0000 or $03FF, the output OC1A/OC1B is held low or high according to the settings of COM1A1/COM1A0 or COM1B1/COM1B0. This is shown in Table 10:

Table 10: PWM Outputs OCR1X = $0000 or $03FF

COM1X1	COM1X0	OCR1X	Output OC1X
1	0	$0000	L
1	0	$03FF	H
1	1	$0000	H
1	1	$03FF	L

Note: X = A or B

In PWM mode, the Timer Overflow Flag1, TOV1, is set when the counter changes direction at $0000. Timer Overflow Interrupt1 operates exactly as in normal Timer/Counter mode, i.e. it is executed when TOV1 is set provided that Timer Overflow Interrupt1 and global interrupts are enabled. This does also apply to the Timer Output Compare1 flags and interrupts.

The PWM output frequency is given by: $f_{PWM} = \frac{f_{TC1}}{2046}$, where f_{TC1} is the Timer/Counter1 Clock Source frequency.

4

The Watchdog Timer

The Watchdog Timer is clocked from a separate on-chip oscillator which runs at 1MHz. By controlling the Watchdog Timer prescaler, the Watchdog reset interval can be adjusted from 16 to 2048 ms. The WDR - Watchdog Reset - instruction resets the Watchdog Timer. From the Watchdog is reset, eight different clock cycle periods can be selected to determine the reset period. If the reset period expires without another Watchdog reset, the AT90S8414 resets and executes from the reset vector.

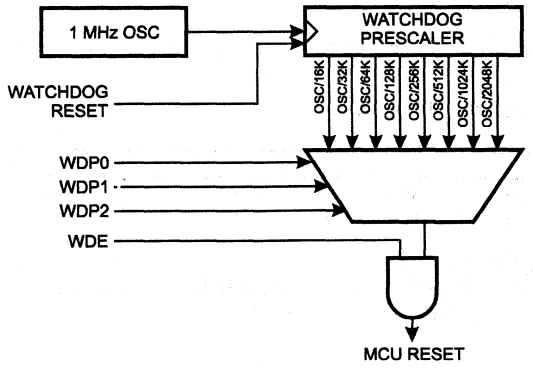


Figure 30: Watchdog Timer

THE WATCHDOG TIMER CONTROL REGISTER - WDTCR

Bit	7	6	5	4	3	2	1	0	
$21	-	-	-	-	WDE	WDP2	WDP1	WDP0	WDTCR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..4 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and will always read as zero.

Bit 3 - WDE : Watch Dog Enable:
When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled.

Bits 2..0 - WDP2, WDP1, WDP0 : Watch Dog Timer Prescaler 2, 1 and 0:
The WDP2, WDP1 and WDP0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is enabled. The different prescaling values and their corresponding Timeout Periods are shown in Table 11.

Table 11: Watch Dog Timer Prescale Select

WDP2	WDP1	WDP0	Timeout Period
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

EEPROM Read/Write Access

The EEPROM access registers are accessible in the I/O space using the IN and OUT instructions.

The write access time is in the range of 2.5 - 4ms, depending on the Vcc voltages. A self-timing function, however, lets the user software detect when the next byte can be written.

The read access time is the same as for the Flash memory and is of no concern to the user software.

THE EEPROM ADDRESS REGISTER - EEAR

Bit	7	6	5	4	3	2	1	0	
$1E	MSB							LSB	EEAR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..0 - EEAR7..0 : EEPROM Address:
The EEPROM Address Register - EEAR7..0 - specifies the EEPROM address in the 256 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 255.

THE EEPROM DATA REGISTER - EEDR

Bit	7	6	5	4	3	2	1	0	
$1D	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..0 - EEDR7..0 : EEPROM Data:
For the EEPROM write operation, the EEDR register contains the data to be written to the EEPROM in the address given by the EEAR register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEAR.

THE EEPROM CONTROL REGISTER - EECR

Bit	7	6	5	4	3	2	1	0	
$1C	-	-	-	-	-	-	EEWE	EERE	EECR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..2 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and will always be read as zero.

Bit 1 - EEWE : EEPROM Write Enable:
The EEPROM Write Enable Signal EEWE is the write strobe to the EEPROM. When address and data are correctly set up, the EEWE bit must be set to write the value into the EEPROM. When the write access time (typically 2.5ms at Vcc=5V or 4ms at Vcc=2.7V) has elapsed, the EEWE bit is cleared (zero) by hardware. The user software can poll this bit and wait for a zero before writing the next byte.

Bit 0 - EERE : EEPROM Read Enable:
The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEAR register, the EERE bit must be set. When the EERE bit is cleared (zero) by hardware, requested data is found in the EEDR register. The EEPROM read access time is within a single clock cycle and there is no need to poll the EERE bit.

The Serial Peripheral Interface - SPI

The Serial Peripheral Interface (SPI) allows high-speed synchronous data transfer between the AT90S8414 and peripheral devices or between several AT90S8414 devices. The AT90S8414 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 5 Mbit/s Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

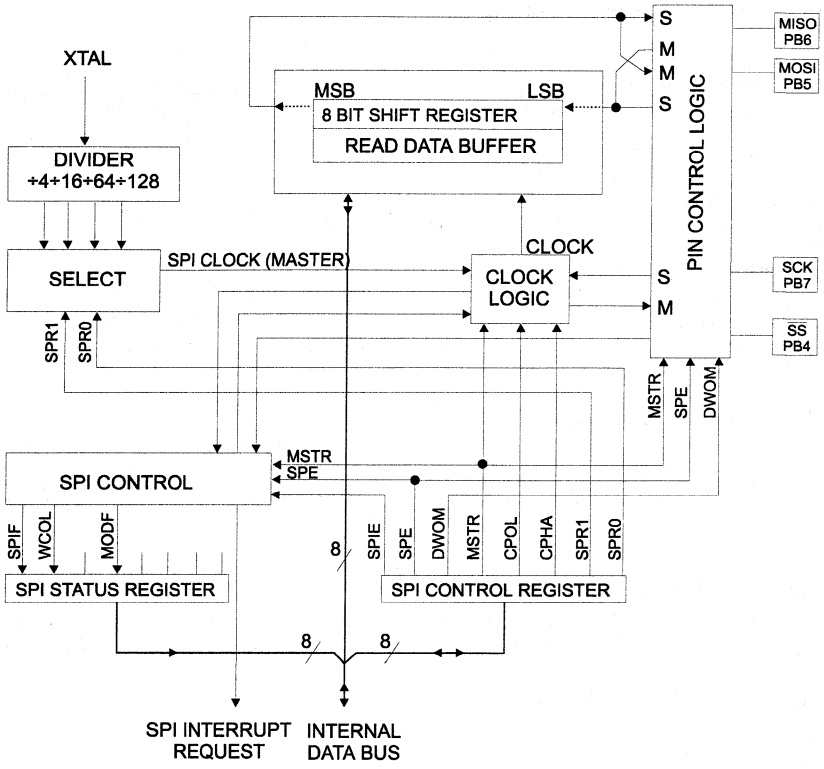


Figure 31: SPI Block Diagram

The interconnection between master and slave CPUs with SPI is shown in Figure 32. The PB7(SCK) pin is the clock output in the master mode and is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the PB5(MOSI) pin and into the PB5(MOSI) pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If the SPI interrupt enable bit (SPIE) in the SPCR register is set, an interrupt is requested. The Slave Select input, PB4($\overline{SS}$), is set low to select an individual SPI device as a slave. When PB4($\overline{SS}$) is set high, the SPI port is deactivated and the PB6(MOSI) pin can be used as an input. Slave/Master mode can also be selected in software by clearing or setting the MSTR bit in the SPI Control Register.

The two shift registers in the Master and the Slave can be considered as one distributed 16-bit circular shift register. This is shown in Figure 32. When data is shifted from the master to the slave, data is also shifted in the opposite direction, simultaneously. This means that during one shift cycle, data in the master and the slave are interchanged.

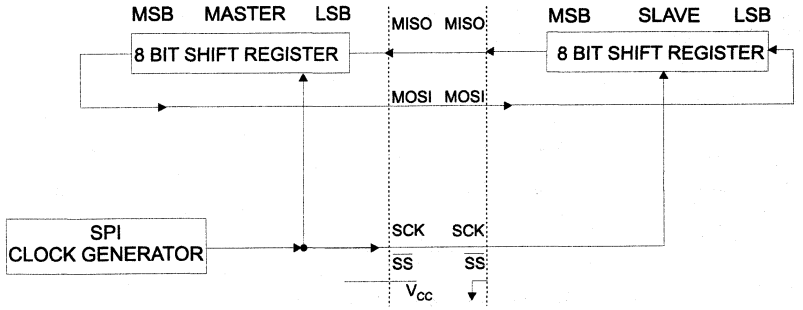


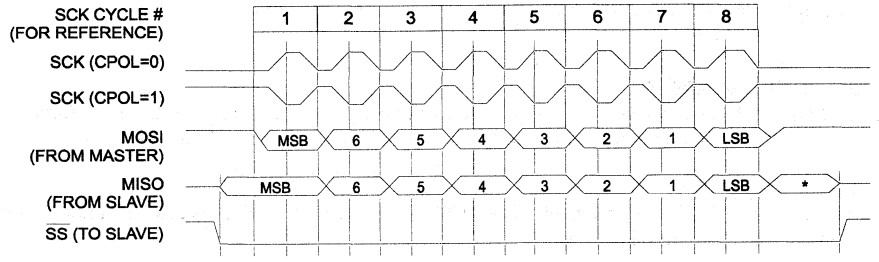
Figure 32: SPI Master-Slave Interconnection

The system is single buffered in the transmit direction and double buffered in the receive direction. This means that characters to be transmitted cannot be written to the SPI Data Register before the entire shift cycle is completed. When receiving data, however, a received character must be read from the SPI Data Register before the next character has been completely shifted in. Otherwise, the first character is lost.

When the SPI is enabled, the DDB5-DDB7 bits in Data Direction Register B (DDRB) are overridden and have no effect. For the PB4($\overline{SS}$) pin, however, DDB4 is assigned the following functionality when the SPI is enabled.

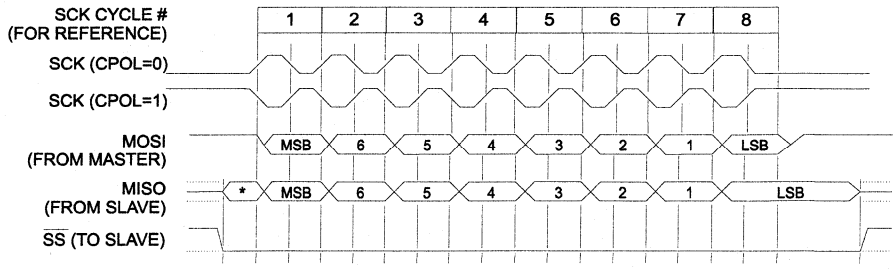
- When DDB4 is cleared (zero) another SPI device can act as master by setting $\overline{SS}$ low. As long as $\overline{SS}$ is held high, the MSTR bit in SPCR selects Master/Slave. If $\overline{SS}$ is set low, MSTR will be forced low.
- When DDB4 is set (one), the PB4($\overline{SS}$) pin can be used as a general output. The MSTR bit selects Master/Slave.

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 33 and Figure 34.



* Not defined but normally MSB of character just received

Figure 33: SPI Transfer Format with CHPA = 0



* Not defined but normally LSB of previously transmitted character

Figure 34: SPI Transfer Format with CHPA = 1

THE SPI CONTROL REGISTER - SPCR

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

Bit 7 - SPIE : SPI Interrupt Enable:
This bit causes setting of the SPIF bit in the SPSR register to execute the SPI interrupt provided that global interrupts are enabled.

Bit 6 - SPE : SPI Enable:
When the SPE bit is set (one), the SPI is enabled and $\overline{SS}$, MOSI, MISO and SCK are connected to pins PB4, PB5, PB6 and PB7.

Bit 5 - DORD : Data ORDer:
When the DORD bit is set (one), the LSB of the data word is transmitted first.
When the DORD bit is cleared (zero), the MSB of the data word is transmitted first.

Bit 4 - MSTR : Master/Slave Select:
This bit selects Master SPI mode when set (one), and Slave SPI mode when cleared (zero). If $\overline{SS}$ on Device1 is set low by Device2 while MSTR is set in Device1, MSTR will be forced low and Device1 will be a slave.

Bit 3 - CPOL : Clock POLarity:
When this bit is set (one), SCK is high when idle. When CPOL is cleared (zero), SCK is low when idle. Refer to Figure 33 and Figure 34 for additional information.

Bit 2 - CPHA : Clock PHAse:
Refer to Figure 33 or Figure 34 for the functionality of this bit.

Bits 1,0 - SPR1, SPR0 : SPI Clock Rate Select 1 and 0:
These two bits control the SCK rate of the device configured as a master. SPR1 and SPR2 have no effect on the slave. The relationship between SCK and the Oscillator Clock frequency f_{cl} is shown in the following table:

Table 12: Relationship Between SCK and the Oscillator Frequency

SPR1	SPR0	SCK Frequency
0	0	$f_{osc}/4$
0	1	$f_{osc}/16$
1	0	$f_{osc}/64$
1	1	$f_{osc}/128$

THE SPI STATUS REGISTER - SPSR

Bit	7	6	5	4	3	2	1	0	
$0E	SPIF	WCOL	-	-	-	-	-	-	SPSR
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - SPIF : SPI Interrupt Flag:
When a serial transfer is complete, the SPIF bit is set (one) and an interrupt is generated if SPIE in SPCR is set (one) and global interrupts are enabled. SPIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the SPIF bit is cleared by first reading the SPI status register with SPIF set (one), then accessing the SPI Data Register (SPDR).

Bit 6 - WCOL : Write COLlision flag:
The WCOL bit is set if the SPI data register (SPDR) is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it will have no effect. The WCOL bit (and the SPIF bit) are cleared (zero) by first reading the SPI Status Register with WCOL set (one), and then accessing the SPI Data Register.

Bit 5..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and will always read as zero.

The SPI interface on the AT90S8414 is also used for program memory and EEPROM downloading or uploading. See Page 4-74 for serial programming and reading.

THE SPI DATA REGISTER - SPDR

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

The SPI Data Register is a read/write register used for data transfer between the register file and the SPI Shift register. Writing to the register initiates data transmission. Reading the register causes the Shift Register Receive buffer to be read.

The UART

The AT90S8414 features a full duplex Universal Asynchronous Receiver and Transmitter (UART). The main features are:

- Baud rate generator generates any baud rate
- High baud rates at low XTAL frequencies
- 8 or 9 bits data
- Noise filtering
- Overrun detection
- Framing Error detection
- False Start Bit detection
- Three separate interrupts on TX Complete, TX Data Register Empty and RX Complete

Data Transmission

A block schematic of the UART transmitter is shown in Figure 35.

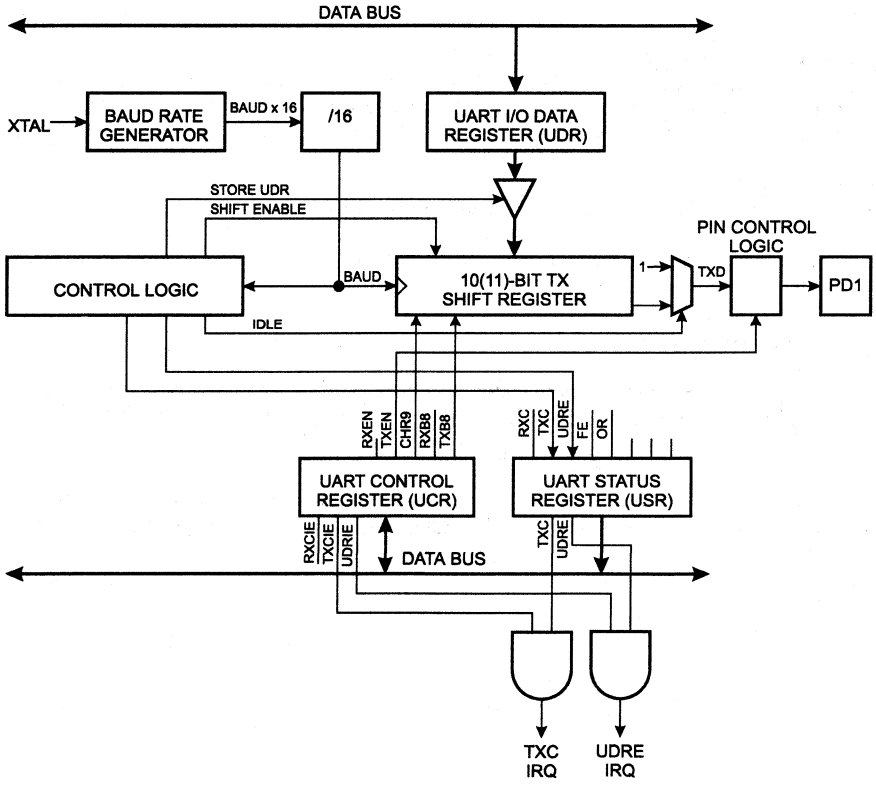


Figure 35: UART Transmitter

Data transmission is initiated by writing the data to be transmitted to the UART I/O Data Register, UDR. Data is transferred from UDR to the Transmit shift register when:

- A new character has been written to UDR after the stop bit from the previous character has been shifted out. The shift register is loaded immediately.
- A new character has been written to UDR before the stop bit from the previous character has been shifted out. The shift register is loaded when the stop bit of the character currently being transmitted has been shifted out.

If the 10(11)-bit Transmitter shift register is empty or when, data is transferred from UDR to the shift register. At this time the UDRE (UART Data Register Empty) bit in the UART Status Register, USR, is set. When this bit is set (one), the UART is ready to receive the next character. At the same time as the data is transferred from UDR to the 10(11)-bit shift register, bit 0 of the shift register is cleared (start bit) and bit 9 or 10 is set (stop bit). If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the TXB8 bit in UCR is transferred to bit 9 in the Transmit shift register.

On the Baud Rate clock following the transfer operation to the shift register, the start bit is shifted out on the TXD pin. Then follows the data, LSB first. When the stop bit has been shifted out, the shift register is loaded if any new data has been written to the UDR during the transmission. During loading, UDRE is set. If there is no new data in the UDR register to send when the stop bit is shifted out, the UDRE flag will remain set. In this case, after the stop bit has been present on TXD for one bit length, the TX Complete Flag, TXC, in USR is set.

The TXEN bit in UCR enables the UART transmitter when set (one). By clearing this bit (zero), the PD1 pin can be used for general I/O. When TXEN is set, the UART Transmitter will be connected to the PD1 pin regardless of the setting of the DDD1 bit in DDRB.

Data Reception

Figure 36 shows a block diagram of the UART Receiver

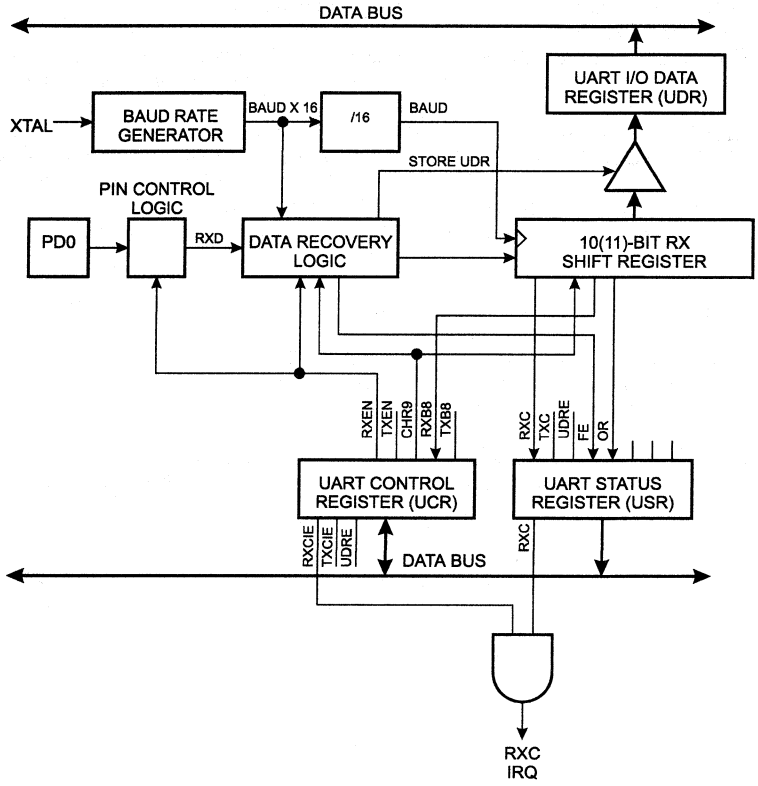


Figure 36: UART Receiver

The receiver front-end logic samples the signal on the RXD pin at a frequency 16 times the baud rate. While the line is idle, one single sample of logical zero will be interpreted as the falling edge of a start bit, and the start bit detection sequence is initiated. Let sample 1 denote the first zero-sample. Following the 1 to 0-transition, the receiver samples the RXD pin at sample 8, 9 and 10. If two or more of these three samples are found to be logical ones, the start bit is rejected as a noise spike and the receiver starts looking for the next 1 to 0-transition.

If however, a valid start bit is detected, sampling of the data bits following the start bit is performed. These bits are also sampled at samples 8, 9 and 10. The logical value found in at least two of the three samples is taken as the bit value. All bits are shifted into the transmitter shift register as they are sampled. Sampling of an incoming character is shown in Figure 37.

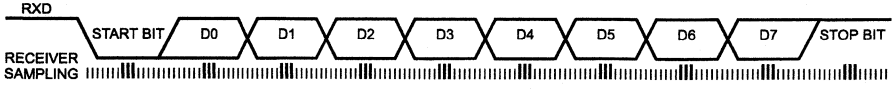


Figure 37: Sampling Received Data

When the stop bit enters the receiver, the majority of the three samples must be one to accept the stop bit. If two or more samples are logical zeros, the Framing Error (FE) flag in the UART Status Register (USR) is set. Before reading the UDR register, the user should always check the FE bit to detect Framing Errors.

Whether or not a valid stop bit is detected at the end of a character reception cycle, the data is transferred to UDR and the RXC flag in USR is set. UDR is in fact two physically separate registers, one for transmitted data and one for received data. When UDR is read, the Receive Data register is accessed, and when UDR is written, the Transmit Data register is accessed. If 9 bit data word is selected (the CHR9 bit in the UART Control Register, UCR is set), the RXB8 bit in UCR is loaded with bit 9 in the Transmit shift register when data is transferred to UDR.

If, after having received a character, the UDR register has not been accessed since the last receive, the OverRun (OR) flag in UCR is set. This means that the new data transferred to the shift register has overwritten the old data not yet read, and the old data is lost. The user should always check the OR bit before reading from the UDR register in order to detect any overruns.

By clearing the RXEN bit in the UCR register, the receiver is disabled. This means that the PD0 pin can be used as a general I/O pin. When RXEN is set, the UART Receiver will be connected to the PD0 pin regardless of the setting of the DDD0 bit in DDRB.

UART Control

THE UART I/O DATA REGISTER - UDR

Bit	7	6	5	4	3	2	1	0	
$0C	MSB							LSB	UDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UDR register is actually two physically separate registers sharing the same I/O address. When writing to the register, the UART Transmit Data register is written. When reading from UDR, the UART Receive Data register is read.

THE UART STATUS REGISTER - USR

Bit	7	6	5	4	3	2	1	0	
$0B	RXC	TXC	UDRE	FE	OR	-	-	-	USR
Read/Write	R	R	R	R	R	R	R	R	
Initial value	0	1	1	0	0	0	0	0	

The USR register is a read-only register providing information on the UART Status.

Bit 7 - RXC: UART Receive Complete:
This bit is set (one) when a received character is transferred from the Receiver Shift register to UDR. The bit is set regardless of any detected framing errors. When the RXCIE bit in UCR is set, setting of RXC causes the UART Receive Complete interrupt to be executed. RXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the bit is cleared (zero) by first reading USR while RXC is set (one) and then reading UDR.

Bit 6 - TXC : UART Transmit Complete:
This bit is set (one) when the entire character (including the stop bit) in the Transmit Shift register has been shifted out and no new data has been written to the UDR. This flag is especially useful in half-duplex communications interfaces, where a transmitting application must enter receive mode and free the communications bus immediately after completing the transmission.

When the TXCIE bit in UCR is set, setting of TXC causes the UART Transmit Complete interrupt to be executed. TXC is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the TXC bit is cleared (zero) by first reading USR while TXC is set and then writing UDR.

This bit is set (one) during reset to indicate that the transmitter is not busy transmitting anything.

Bit 5 - UDRE : UART Data Register Empty:
This bit is set (one) when a character written to UDR is transferred to the Transmit shift register. Setting of this bit indicates that the transmitter is ready to receive a new character for transmission.

When the UDRIE bit in UCR is set, setting of UDRE causes the UART Transmit Complete interrupt to be executed. UDRE is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, the UDRE bit is cleared (zero) by first reading USR while UDRE is set and then writing UDR.

UDRE is set (one) during reset to indicate that the transmitter is ready.

Bit 4 - FE : Framing Error:
This bit is set if a Framing Error condition is detected, i.e. when the stop bit of an incoming character is zero.

The FE bit is cleared (zero) by first reading USR while FE is set and then reading UDR.

Bit 3 - OR : OverRun:
This bit is set if an Overrun condition is detected, i.e. when a character already present in the UDR register is not read before the next character is transferred from the Receiver Shift register.

The OR bit is cleared (zero) by first reading USR while OR is set and then reading UDR.

Bits 2..0 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and will always read as zero.

THE UART CONTROL REGISTER - UCR

Bit	7	6	5	4	3	2	1	0	
$0A	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	UCR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	
Initial value	0	0	0	0	0	0	0	0	

Bit 7 - RXCIE : RX Complete Interrupt Enable:
When this bit is set (one), a setting of the RXC bit in USR will cause the Receive Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 6 - TXCIE : TX Complete Interrupt Enable:
When this bit is set (one), a setting of the TXC bit in USR will cause the Transmit Complete interrupt routine to be executed provided that global interrupts are enabled.

Bit 5 - UDRIE : UART Data Register Empty Interrupt Enable:
When this bit is set (one), a setting of the UDRE bit in USR will cause the UART Data Register Empty interrupt routine to be executed provided that global interrupts are enabled.

Bit 4 - RXEN : Receiver Enable:
This bit enables the UART receiver when set (one). When the receiver is disabled, the TXC, OR and FE status flags cannot become set. If these flags are set, turning off RXEN does not cause them to be cleared.

Bit 3 - TXEN : Transmitter Enable:
This bit enables the UART transmitter when set (one). When disabling the transmitter while transmitting a character, the transmitter is not disabled before the character in the shift register plus any following character in UDR has been completely transmitted.

Bit 2 - CHR9 : 9 Bit Characters: When this bit is set (one) transmitted and received characters are 9 bit long plus start and stop bits. The 9th bit is read and written by using the RXB8 and TXB8 bits in UCR, respectively. The 9th data bit can be used as an extra stop bit or a parity bit.

Bit 1 - RXB8 : Receive Data Bit 8
When CHR9 is set (one), RXB8 is the 9th data bit of the received character.

Bit 0 - TXB8 : Transmit Data Bit 8
When CHR9 is set (one), TXB8 is the 9th data bit in the character to be transmitted.

THE BAUD RATE GENERATOR

The baud rate generator is a frequency divider which generates baud-rates according to the following equation:

$$BAUD = \frac{f_{ck}}{16(UBRR + 1)}$$

- BAUD = Baud-Rate
- f_{ck} = Crystal Clock frequency
- UBRR = Contents of the UART Baud Rate register, UBRR (0-255)

For standard crystal frequencies, the most commonly used baud rates can be generated by using the UBRR settings in Table 13. UBRR values which yield an actual baud rate differing less than 2% from the target baud rate, are bolded in the table.

Table 13: UBRR Settings at Various Crystal Frequencies

Baud Rate	1 MHz		%Error	1.8432 MHz		%Error	2 MHz		%Error	2.4576 MHz		%Error
2400	UBRR=	25	0.2	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	63	0.0
4800	UBRR=	12	0.2	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	31	0.0
9600	UBRR=	6	7.5	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	15	0.0
14400	UBRR=	3	7.8	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	10	3.1
19200	UBRR=	2	7.8	UBRR=	5	0.0	UBRR=	6	7.5	UBRR=	7	0.0
28800	UBRR=	1	7.8	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	6.3
57600	UBRR=	0	7.8	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	12.5
115200	UBRR=	0	84.3	UBRR=	0	0.0	UBRR=	0	7.8	UBRR=	0	25.0

Baud Rate	3.2768 MHz		%Error	3.6864 MHz		%Error	4 MHz		%Error	4.608 MHz		%Error
2400	UBRR=	84	0.4	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0
4800	UBRR=	42	0.8	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0
9600	UBRR=	20	1.6	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0
14400	UBRR=	13	1.6	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0
19200	UBRR=	10	3.1	UBRR=	11	0.0	UBRR=	12	0.2	UBRR=	14	0.0
28800	UBRR=	6	1.6	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0
57600	UBRR=	3	12.5	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0
115200	UBRR=	1	12.5	UBRR=	1	0.0	UBRR=	1	7.8	UBRR=	2	20.0

Baud Rate	7.3728 MHz		%Error	8 MHz		%Error	9.216 MHz		%Error	11.059 MHz		%Error
2400	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	287	-
4800	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	143	0.0
9600	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	71	0.0
14400	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	47	0.0
19200	UBRR=	23	0.0	UBRR=	25	0.2	UBRR=	29	0.0	UBRR=	35	0.0
28800	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	23	0.0
57600	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	11	0.0
115200	UBRR=	3	0.0	UBRR=	3	7.8	UBRR=	4	0.0	UBRR=	5	0.0

Baud Rate	14.746 MHz		%Error	16 MHz		%Error	18.432 MHz		%Error	20 MHz		%Error
2400	UBRR=	383	-	UBRR=	416	-	UBRR=	479	-	UBRR=	520	-
4800	UBRR=	191	0.0	UBRR=	207	0.2	UBRR=	239	0.0	UBRR=	259	-
9600	UBRR=	95	0.0	UBRR=	103	0.2	UBRR=	119	0.0	UBRR=	129	0.2
14400	UBRR=	63	0.0	UBRR=	68	0.6	UBRR=	79	0.0	UBRR=	86	0.2
19200	UBRR=	47	0.0	UBRR=	51	0.2	UBRR=	59	0.0	UBRR=	64	0.2
28800	UBRR=	31	0.0	UBRR=	34	0.8	UBRR=	39	0.0	UBRR=	42	0.9
57600	UBRR=	15	0.0	UBRR=	16	2.1	UBRR=	19	0.0	UBRR=	21	1.4
115200	UBRR=	7	0.0	UBRR=	8	3.7	UBRR=	9	0.0	UBRR=	10	1.4

THE UART BAUD RATE REGISTER - UBRR

Bit	7	6	5	4	3	2	1	0	
$09	MSB							LSB	UBRR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

The UBRR register is an 8-bit read/write register which specifies the UART Baud Rate according to the description on Page 4-51.

The Analog Comparator

The analog comparator compares the input values on the positive pin PB2 (AIN0) and negative pin PB3 (AIN1). When the voltage on the positive pin PB2 (AIN0) is higher than the voltage on the negative pin PB3 (AIN1), the Analog Comparator Output, ACO is set (one). The comparator's output can be set to trigger the Timer/Counter1 Input Capture function. In addition, the comparator can trigger a separate interrupt, exclusive to the Analog Comparator. The user can

select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 38.

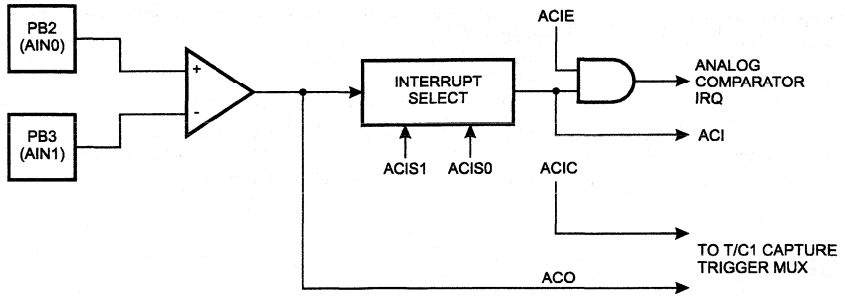


Figure 38: Analog Comparator Block Diagram

4

THE ANALOG COMPARATOR CONTROL AND STATUS REGISTER - ACSR

Bit	7	6	5	4	3	2	1	0	
$08	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	ACSR
Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

Bits 7..6 - Res : Reserved bits:
These bits are reserved bits in the AT90S8414 and will always read as zero.

Bit 5 - ACO : Analog Comparator Output:
ACO is directly connected to the comparator output.

Bit 4 - ACI : Analog Comparator Interrupt Flag:
This bit is set (one) when a comparator output event triggers the interrupt mode defined by ACI1 and ACI0. The Analog Comparator Interrupt routine is executed if the ACIE bit is set (one) and the I-bit in SREG is set (one). ACI is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, ACI is cleared by writing a logic one to the flag.

Bit 3 - ACIE : Analog Comparator Interrupt Enable:
When the ACIE bit is set (one) and the I-bit in the Status Register is set (one), the analog comparator interrupt is activated. When cleared (zero), the interrupt is disabled. For details on the comparator, refer to Page 4-52.

Bit 2 - ACIC : Analog Comparator Input Capture enable:
When set (one), this bit enables the Input Capture function in Timer/Counter1 to be triggered by the analog comparator. The comparator output is in this case directly connected to the Input Capture front-end logic, making the comparator utilize the noise canceler and edge select features of the Timer/Counter1 Input Capture interrupt. When cleared (zero), no connection between the analog comparator and the Input Capture function is given. To make the comparator trigger the Timer/Counter1 Input Capture interrupt, the TICIE1 bit in the Timer Interrupt Mask Register (TIMSK) must be set (one).

Bits 1,0 - ACIS1, ACIS0 : Analog Comparator Interrupt Mode Select:
These bits determine which comparator events that trigger the Analog Comparator interrupt. The different settings are shown in
Table 14.

Table 14: ACIS1/ACIS0 Settings

ACIS1	ACIS0	Interrupt Mode
0	0	Comparator Interrupt on Output Toggle
0	1	Reserved
1	0	Comparator Interrupt on Falling Output Edge
1	1	Comparator Interrupt on Rising Output Edge

Note: When changing the ACIS1/ACIS0 bits, The Analog Comparator Interrupt must be disabled by clearing its Interrupt Enable bit in the ACSR register. Otherwise an interrupt can occur when the bits are changed.

I/O-Ports

Port A

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

Three data memory address locations are allocated for the Port A, one each for the Data Register - PORTA ($1B), Data Direction Register - DDRA ($1A) and the Port A Input Pins - PINA ($19). The Port A Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pullups. The PORT A output buffers can sink 20mA and thus drive LED displays directly. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

The PORT A pins have alternate functions related to the optional external data SRAM. PORT A can be configured to be the multiplexed low-order address/data bus during accesses to the external data memory. In this mode, PORT A has internal pullups.

PORT A also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

When PORT A is set to the alternate function by the SRE - External SRAM Enable - bit in the MCUCR - MCU Control Register, the alternate settings override the data direction register.

THE PORT A DATA REGISTER - PORTA

Bit	7	6	5	4	3	2	1	0	
$1B	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	**PORTA**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT A DATA DIRECTION REGISTER - DDRA

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

THE PORT A INPUT PINS ADDRESS - PINA

Bit	7	6	5	4	3	2	1	0	
$19	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	**PINA**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port A Input Pins address - PINA - is not a register, and this address enables access to the physical value on each Port A pin. When reading PORTA the PORTA Data Latch is read, and when reading PINA, the logical values present on the pins are read.

PORTA AS GENERAL DIGITAL I/O

All 8 bits in PORT A are equal when used as digital I/O pins.

PAn, General I/O pin: The DDAn bit in the DDRA register selects the direction of this pin, if DDAn is set (one), PAn is configured as an output pin. If DDAn is cleared (zero), PAn is configured as an input pin. If PAn is set (one) when the pin configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PAn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 15: DDAn Effects on PORT A Pins

DDAn	PAn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6...0, pin number.

PORT A SCHEMATICS

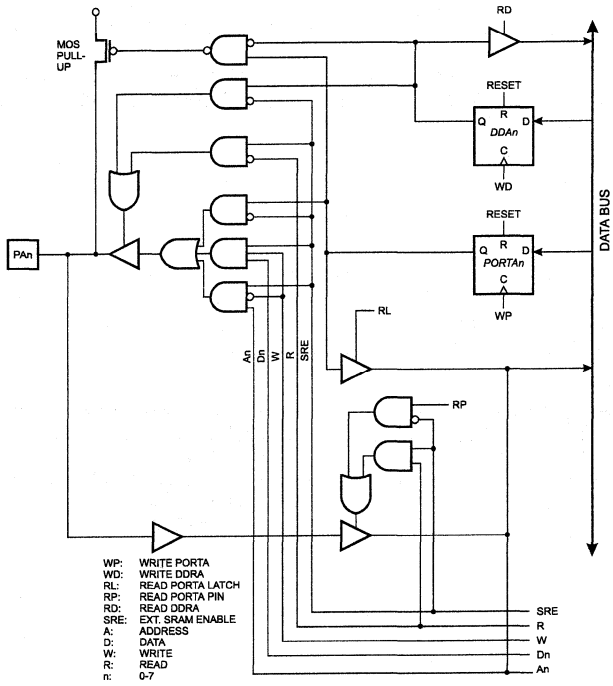


Figure 39: PORTA Schematic Diagrams (Pins PA0 - PA7)

Port B

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

Three data memory address locations are allocated for the Port B, one each for the Data Register - PORTB ($18), Data Direction Register - DDRB ($17) and the Port B Input Pins - PINB ($16). The Port B Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pullups. The Port B output buffers can sink 20mA and thus drive LED displays directly. When pins PB0 to PB7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

The Port B pins with alternate functions are shown in the following table:

Table 16: Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB0	T0 (Timer/Counter 0 external counter input)
PB1	T1 (Timer/Counter 1 external counter input)
PB2	AIN0 (Analog comparator positive input)
PB3	AIN1 (Analog comparator negative input)
PB4	$\overline{SS}$ (SPI Slave Select input)
PB5	MOSI (SPI Bus Master Output/Slave Input)
PB6	MISO (SPI Bus Master Input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

When the pins are used for the alternate function the DDRB and PORTB register has to be set according to the alternate function description.

THE PORT B DATA REGISTER - PORTB

Bit	7	6	5	4	3	2	1	0	
$18	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	**PORTB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B DATA DIRECTION REGISTER - DDRB

Bit	7	6	5	4	3	2	1	0	
$17	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	**DDRB**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT B INPUT PINS ADDRESS - PINB

Bit	7	6	5	4	3	2	1	0	
$16	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	**PINB**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port B Input Pins address - PINB - is not a register, and this address enables access to the physical value on each Port B pin. When reading PORTB, the PORTB Data Latch is read, and when reading PINB, the logical values present on the pins are read.

PORTB AS GENERAL DIGITAL I/O

All 8 bits in port B are equal when used as digital I/O pins.

PBn, General I/O pin: The DDBn bit in the DDRB register selects the direction of this pin, if DDBn is set (one), PBn is configured as an output pin. If DDBn is cleared (zero), PBn is configured as an input pin. If PBn is set (one) when the pin configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PBn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 17: DDBn Effects on Port B Pins

DDBn	PBn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PBn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTB

The alternate pin configuration is as follows:

SCK - PORTB, Bit 7:

SCK: Master clock output, slave clock input pin for SPI channel

MISO - PORTB, Bit 6:

MISO: Master data input, slave data output pin for SPI channel

MOSI - PORTB, Bit 5:

MOSI: SPI Master data output, slave data input for SPI channel

$\overline{SS}$ - PORTB, Bit 4:

$\overline{SS}$ (Slave port select input)

AIN1 - PORTB, Bit 3

AIN1, Analog Comparator Negative Input. When configured as an input (DDB3 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB3 is cleared (zero)), this pin also serves as the negative input of the on-chip analog comparator.

AIN0 - PORTB, Bit 2

AIN0, Analog Comparator Positive Input. When configured as an input (DDB2 is cleared (zero)) and with the internal MOS pull up resistor switched off (PB2 is cleared (zero)), this pin also serves as the positive input of the on-chip analog comparator.

T1 - PORTB, Bit 1:

T1, Timer/Counter1 counter source: The PB1 pin has to be configured as an input (DDB1 is cleared (zero)) to serve this function. See the timer description for further details. The internal pull up MOS resistor can be activated as described above.

T0 - PORTB, Bit 0:

T0: Timer/Counter0 counter source: The PB0 pin has to be configured as an input (DDB0 is cleared (zero)) to serve this function. See the timer description for further details. The internal pull up MOS resistor can be activated as described above.

PORT B SCHEMATICS

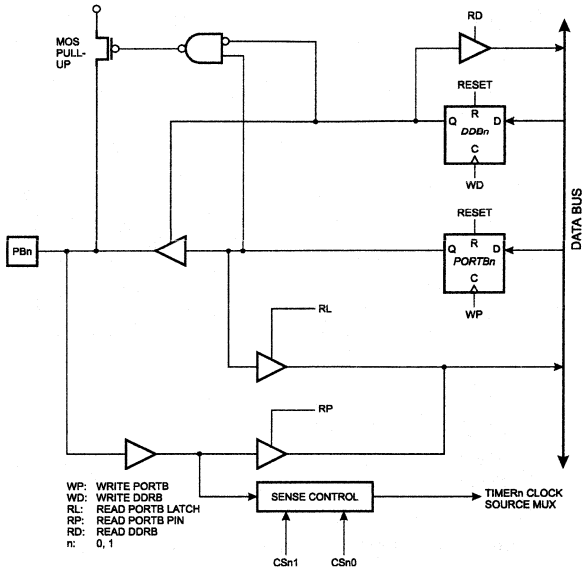


Figure 40: PORTB Schematic Diagram (Pins PB0 and PB1)

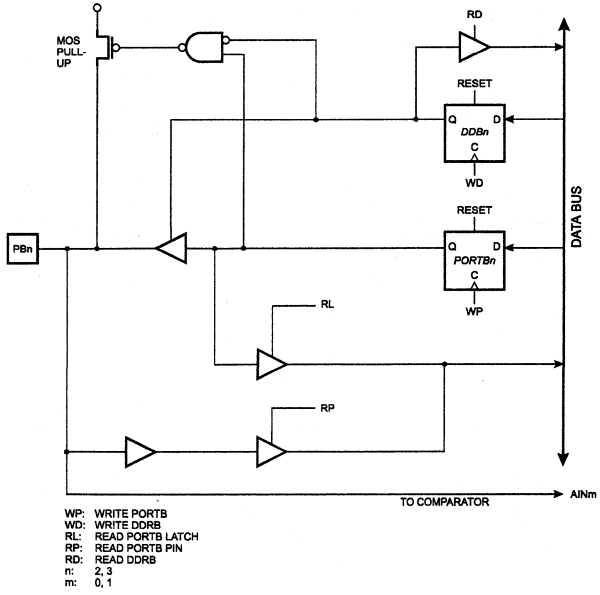


Figure 41: PORTB Schematic Diagram (Pins PB2 and PB3)

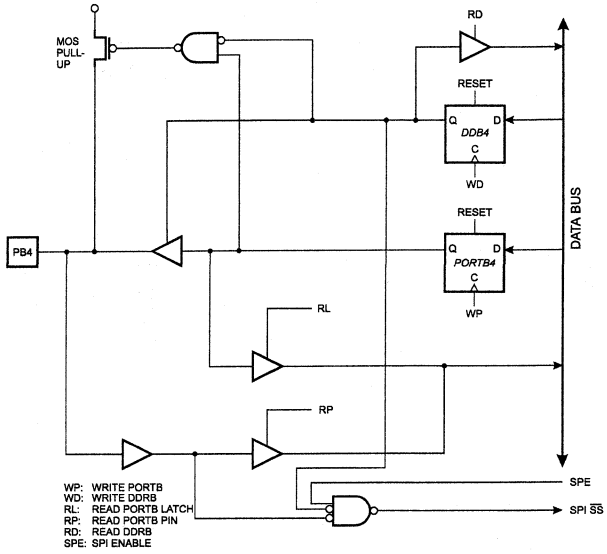


Figure 42: PORTB Schematic Diagram (Pin PB4)

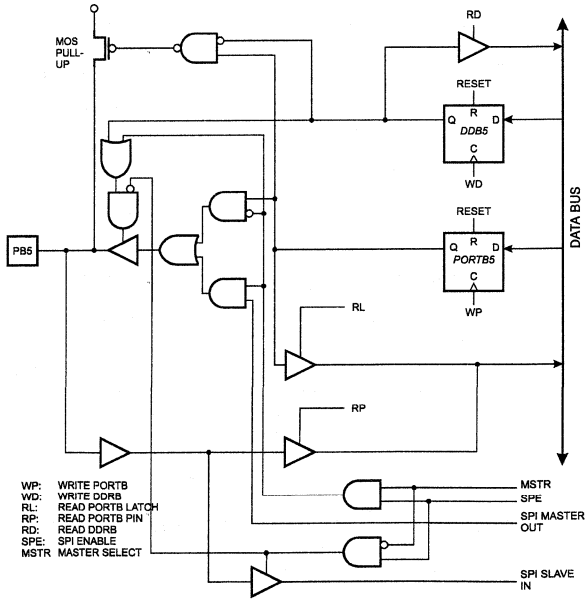


Figure 43: PORTB Schematic Diagram (Pin PB5)

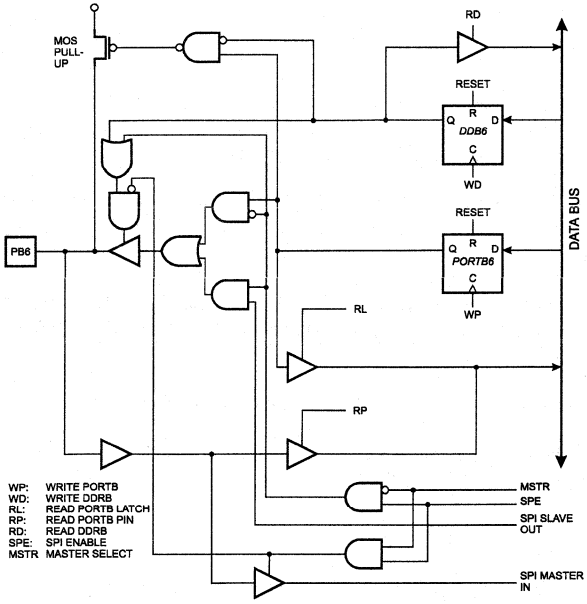


Figure 44: PORTB Schematic Diagram (Pin PB6)

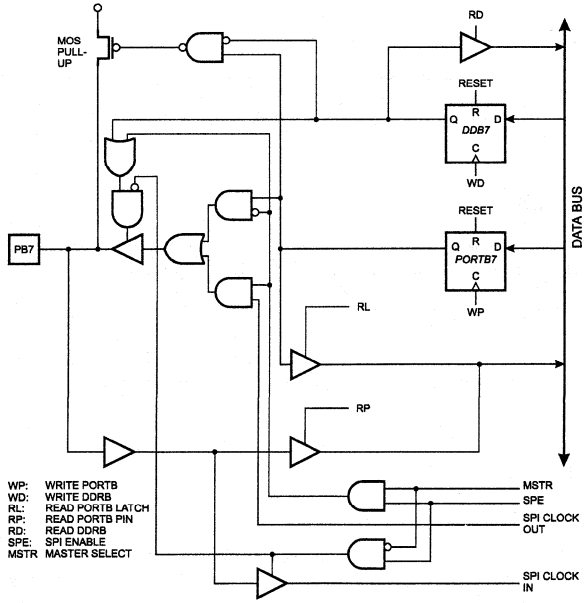


Figure 45: PORTB Schematic Diagram (Pin PB7)

Port C

PORT C is an 8-bit bi-directional I/O port.

Three data memory address locations are allocated for the Port C, one each for the Data Register - PORTC ($15), Data Direction Register - DDRC ($14) and the Port C Input Pins - PINC ($13). The Port C Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

All port pins have individually selectable pullups. The PORT C output buffers can sink 20mA and thus drive LED displays directly. When pins PC0 to PC7 are used as inputs and are externally pulled low, they will source current (IIL) if the internal pullups are activated.

The PORT C pins have alternate functions related to the optional external data SRAM. PORT C can be configured to be the high-order address byte during accesses to external data memory. In this mode, PORT C uses internal pullups when emitting 1's.

PORT C also receives the high-order address bits and some control signals during Flash programming and verification.

When PORT C is set to the alternate function by the SRE - External SRAM Enable - bit in the MCUCR - MCU Control Register, the alternate settings override the data direction register.

THE PORT C DATA REGISTER - PORTC

Bit	7	6	5	4	3	2	1	0	
$15	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	**PORTC**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT C DATA DIRECTION REGISTER - DDRC

Bit	7	6	5	4	3	2	1	0	
$14	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	**DDRC**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT C INPUT PINS ADDRESS - PINC

Bit	7	6	5	4	3	2	1	0	
$13	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	**PINC**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port C Input Pins address - PINC - is not a register, and this address enables access to the physical value on each Port C pin. When reading PORTC, the PORTC Data Latch is read, and when reading PINC, the logical values present on the pins are read.

PORTC AS GENERAL DIGITAL I/O

All 8 bits in PORT C are equal when used as digital I/O pins.

PCn, General I/O pin: The DDCn bit in the DDRC register selects the direction of this pin, if DDCn is set (one), PCn is configured as an output pin. If DDCn is cleared (zero), PCn is configured as an input pin. If PCn is set (one) when the pin configured as an input pin, the MOS pull up resistor is activated. To switch the pull up resistor off, the PCn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 18: DDCn Effects on PORT C Pins

DDCn	PCn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PAn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7...0, pin number.

PORT C SCHEMATICS

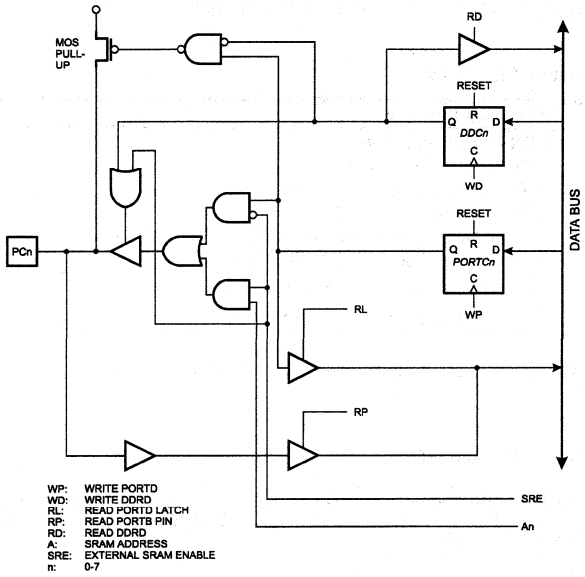


Figure 46: PORTC Schematic Diagram (Pins PC0 - PC7)

Port D

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

Three data memory address locations are allocated for the Port D, one each for the Data Register - PORTD ($12), Data Direction Register - DDRD ($11) and the Port D Input Pins - PIND ($10). The Port D Input Pins address is read only, while the Data Register and the Data Direction Register are read/write.

The Port D output buffers can sink 20 mA. As inputs, Port D pins that are externally pulled low will source current (IIL) if the pullups are activated.

Some Port D pins have alternate functions as shown in the following table:

Table 19: Port D Pins Alternate Functions

Port Pin	Alternate Function
PD0	RDX (UART Input line)
PD1	TDX (UART Output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1 (External interrupt 1 input)
PD4	OC0 (Timer/Counter0 Output compare match output)
PD5	OC1A (Timer/Counter1 Output compareA match output)
PD6	$\overline{WR}$ (Write strobe to external memory)
PD7	$\overline{RD}$ (Read strobe to external memory)

When the pins are used for the alternate function the DDRD and PORTD register has to be set according to the alternate function description.

THE PORT D DATA REGISTER - PORTD

Bit	7	6	5	4	3	2	1	0	
$12	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	**PORTD**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D DATA DIRECTION REGISTER - DDRD

Bit	7	6	5	4	3	2	1	0	
$11	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	**DDRD**
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	0	0	

THE PORT D INPUT PINS ADDRESS - PIND

Bit	7	6	5	4	3	2	1	0	
$10	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	**PIND**
Read/Write	R	R	R	R	R	R	R	R	
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

The Port D Input Pins address - PIND - is not a register, and this address enables access to the physical value on each Port D pin. When reading PORTD, the PORTD Data Latch is read, and when reading PIND, the logical values present on the pins are read.

PORTD AS GENERAL DIGITAL I/O

PDn, General I/O pin: The DDDn bit in the DDRD register selects the direction of this pin. If DDDn is set (one), PDn is configured as an output pin. If DDDn is cleared (zero), PDn is configured as an input pin. If PDn is set (one) when configured as an input pin the MOS pull up resistor is activated. To switch the pull up resistor off the PDn has to be cleared (zero) or the pin has to be configured as an output pin.

Table 20: DDDn Bits on Port D Pins

DDDn	PDn	I/O	Pull up	Comment
0	0	Input	No	Tri-state (Hi-Z)
0	1	Input	Yes	PDn will source current (IIL) if ext. pulled low.
1	0	Output	No	Push-Pull Zero Output
1	1	Output	No	Push-Pull One Output

n: 7,6...0, pin number.

ALTERNATE FUNCTIONS FOR PORTD

$\overline{RD}$ - PORTD, Bit 7:

$\overline{RD}$ is the external data memory read control strobe.

$\overline{WR}$ - PORTD, Bit 6:

$\overline{WR}$ is the external data memory write control strobe.

OC1- PORTD, Bit 5:

OC1, Output compare match output: The PD5 pin can serve as an external output when the Timer/Counter1 compare matches. The PD5 pin has to be configured as an output (DDD5 set (one)) to serve this function. See the Timer/Counter1 description for further details, and how to enable the output. The OC1 pin is also the output pin for the PWM mode timer function.

OC0- PORTD, Bit 4:

OC0, Output compare match output: The PD4 pin can serve as an external output when the Timer/Counter0 compare matches. The PD4 pin has to be configured as an output (DDD4 set (one)) to serve this function. See the Timer/Counter0 description for further details, and how to enable the output.

INT1 - PORTD, Bit 3:

INT1, External Interrupt source 1: The PD3 pin can serve as an external active low interrupt source to the MCU. The PD3 pin has to be configured as an input (DDD3 is cleared (zero)) to serve this function. The internal pull up MOS resistor can be activated as described above. See the interrupt description for further details, and how to enable the source.

INT0 - PORTD, Bit 2:

INT0, External Interrupt source 0: The PD2 pin can serve as an external active low interrupt source to the MCU. The PD2 pin has to be configured as an input (DDD2 is cleared (zero)) to serve this function. The internal pull up MOS resistor can be activated as described above. See the interrupt description for further details, and how to enable the source.

TXD - PORTD, Bit 1:

TXD is the serial UART output pin. See "The UART".

RXD - PORTD, Bit 0:

RXD is the serial UART output pin. See "The UART".

PORTD SCHEMATICS

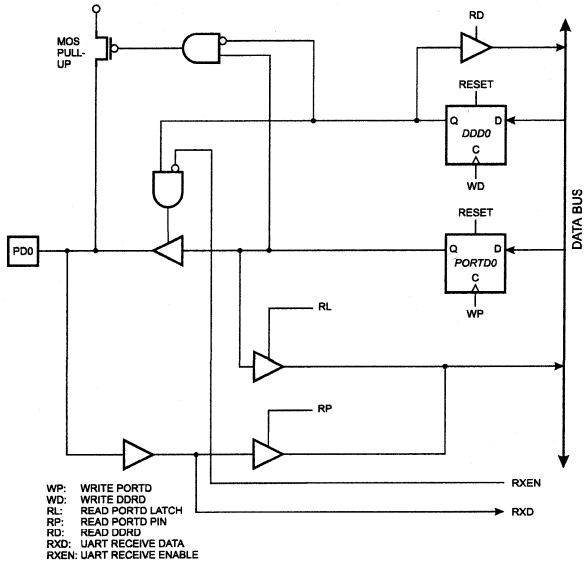


Figure 47: PORTD Schematic Diagram (Pin PD0)

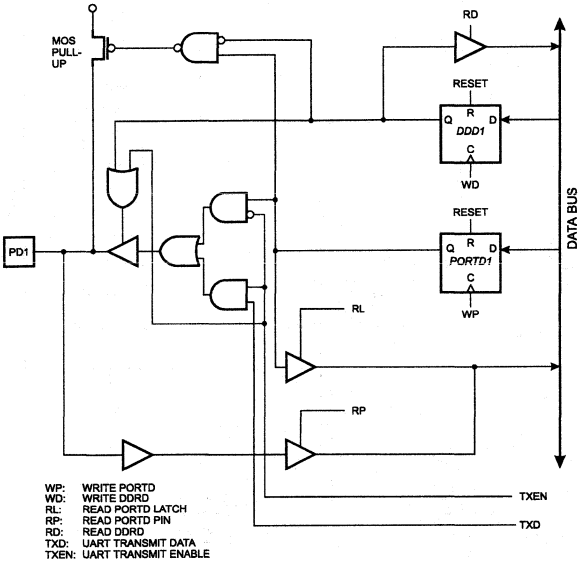


Figure 48: PORTD Schematic Diagram (Pin PD1)

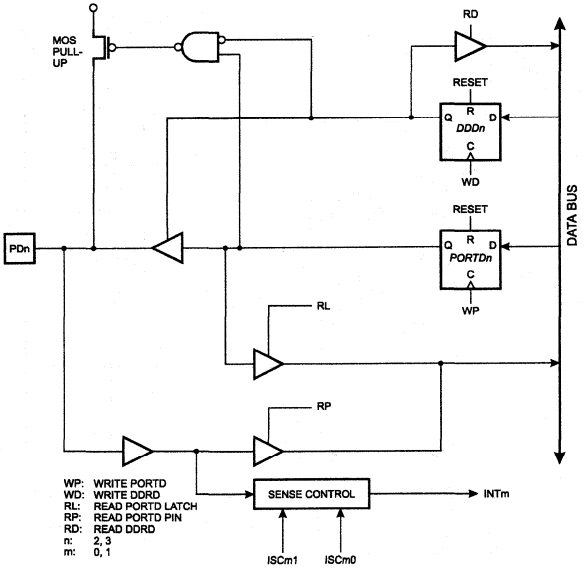


Figure 49: PORTD Schematic Diagram (Pins PD2 and PD3)

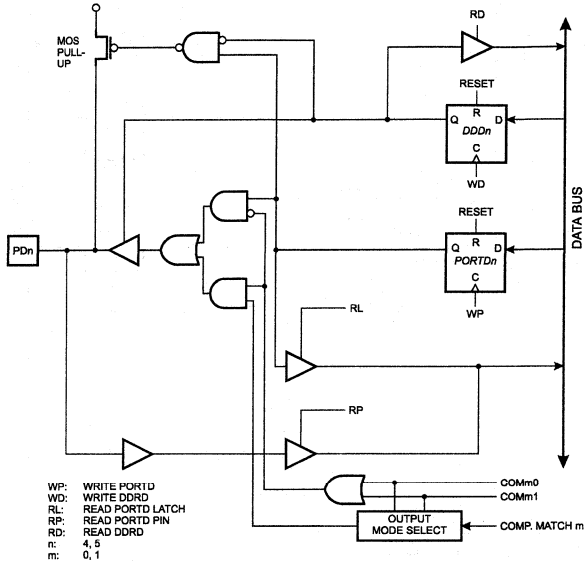


Figure 50: PORTD Schematic Diagrams (Pin PD4 and PD5)

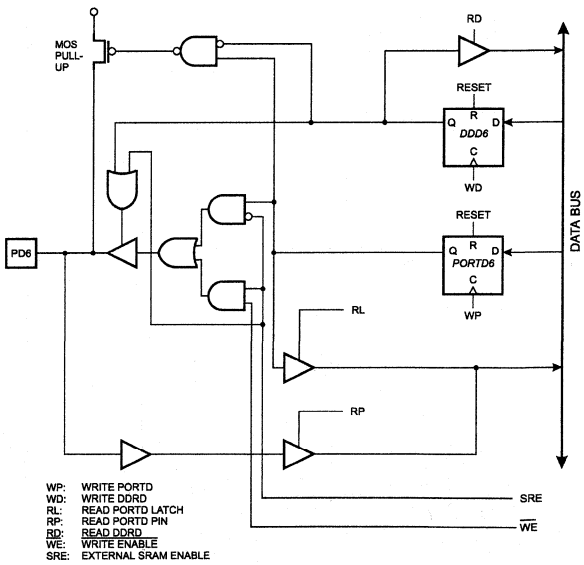


Figure 51: PORTD Schematic Diagram (Pin PD6)

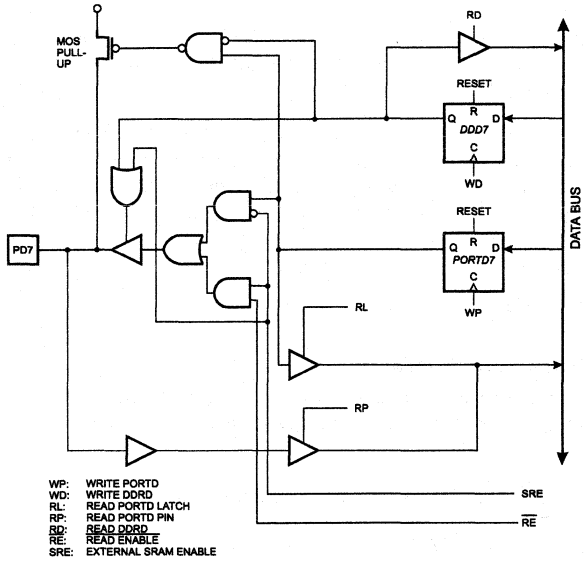


Figure 52: PORTD Schematic Diagram (Pin PD7)

Memory Programming

Program Memory Lock Bits

The AT90S8414 MCU provides three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in Table 21.

Table 21: Lock Bit Protection Modes

Program Lock Bits			Protection Type
Mode	LB1	LB2	
1	U	U	No program lock features
2	P	U	Further programming of the Flash is disabled
3	P	P	Same as mode 2, but verify is also disabled.

Note: The Lock Bits can only be erased with the Chip Erase operation.

Programming the Flash and EEPROM

Atmel's AT90S8414 offers 8K bytes of in-system reprogrammable Flash PEROM Program memory and 256 bytes of EEPROM Data memory.

The AT90S8414 is normally shipped with the on-chip PEROM Program and EEPROM Data memory arrays in the erased state (i.e. contents = $FF) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage Serial programming mode. The serial programming mode provides a convenient

way to download the Program and Data into the AT90S8414 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers.

The Program and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one continuos address space: $0000 to $1FFF for the Program array and $2000 to $20FF for the Data array.

The Program and Data memory arrays on the AT90S8414 are programmed byte-by-byte in either programming modes. An auto-erase cycle is provided with the self-timed programming operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode. In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

Parallel Programming

PARALLEL PROGRAMMING ALGORITHM

All major programming vendors offer worldwide support for the Atmel microcontroller series. To program and verify the AT90S8414 in the parallel programming mode, the following sequence is recommended:

1. *Power-up sequence:*
   Apply power between VCC and GND pins with all other pins floating.
   Set RST pin to 'H'.
   Apply a 1 MHz to 20 MHz clock to XTAL1 pin and wait for at least 10 ms.
2. Set ALE pin to 'H'.
   Set ICP pin to 'H' and all other pins to 'H'.
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins PC6, PC7, PD6, PD7 to select one of the programming operations shown in Table 22.
4. *Programming and verifying:*
   Apply the desired byte address to pins PB0 - PB7 and PC0 - PC5.
   Apply data to pins PA0 - PA7 for Write Program operation.
5. Raise ICP/VPP to 12 V to enable Flash programming, erase or verification.
6. Pulse ALE/$\overline{\text{PROG}}$ once to program a byte in the Program memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.
7. To verify the byte just programmed, bring pin PC7 to 'L' and read the programmed data at pins PA0 - PA7.
8. Repeat steps 3 through 7 changing the address and data for the entire 256 or 8K bytes array or until the end of the object file is reached.
9. *Power-off sequence:*
   Set XTAL1 to 'L'.
   Set RST and ICP/VPP pins to 'L'.
   Float all other I/O pins.
   Turn VCC power off.

In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

$\overline{\text{DATA}}$ POLLING

The AT90S8414 features $\overline{\text{DATA}}$ Polling to indicate the end of a write cycle. During a write cycle in the parallel programming mode, an attempted read of the last byte written will result in the complement of the written datum on PA7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{\text{DATA}}$ Polling may begin any time after a write cycle has been initiated.

READY/$\overline{\text{BUSY}}$

The progress of byte programming in the parallel programming mode can also be monitored by the RDY/$\overline{\text{BSY}}$ output signal. Pin PD4 is pulled Low after ALE goes High during programming to indicate $\overline{\text{BUSY}}$. PD4 is pulled High again when programming is done to indicate READY.

PROGRAM VERIFY

If lock bits LB1 and LB2 have not been programmed, Program data can be read back via the data lines for verification:

1. Reset the internal address counter to $000 by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Program data and read the output data at the port PB pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next Program data byte at the port PB pins.
5. Repeat steps 3 and 4 until the entire array is read. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

CHIP ERASE

Both Flash and EEPROM arrays are erased electrically at the same time. In parallel programming mode, chip erase is initiated by using the proper combinations of control signals and by holding ALE/$\overline{\text{PROG}}$ low for 10 ms. The Program and Data arrays are written with all '1's in the Chip Erase operation.

In the serial programming mode, the chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and take about 16 ms.

During chip erase, a serial read from any address location will return $00 at the data outputs.

SERIAL PROGRAMMING FUSE

A programmable fuse is available to disable Serial Programming if the user needs maximum security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode.

The AT90S8414 is shipped with the Serial Programming Mode enabled.

READING THE SIGNATURE BYTES

The signature bytes are read by the same procedure as a normal verification of locations $000 ,$001 and $000, except that PD6 and PD7 must be pulled to a logic low. The values returned are as follows:
($000) = $1E indicates manufactured by Atmel.
($001) = $93 indicates 8K bytes Program Flash Memory.
($002) = $01 indicates 90S8414 device when ($001) = $93.

Table 22: Flash and EEPROM Parallel Programming Modes

Mode	RST	ALE/ $\overline{PROG}$	ICP/ Vpp	PC6	PC7	PD6	PD7	Data I/O PA7:0	Addr PC5:0, PB7:0
Chip Erase	H	(1)	12V	H	L	L	L	X	X
Write (8.25K Bytes) Memory	H		12V	L	H	H	H	DIN	ADDR
Read (8.25 K Bytes) Memory	H	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bit 1	H		12V	H	L	H	L	PA7=0	X
Write Lock Bit 2	H		12V	H	H	L	L	PA6=0	X
Read Lock Bit 1	H	H	12V	H	H	L	L	@PA1	X
Read Lock Bit 2	H	H	12V	H	H	L	L	@PA0	X
Read Atmel Code	H	H	12V	L	L	L	L	DOUT	$30
Read Device Code	H	H	12V	L	L	L	L	DOUT	$31
Serial Prog. Enable	H	(1)	12V	L	H	L	H	PA0=0	X
Serial Prog. Disable	H	(1)	12V	L	H	L	H	PA0=1	X
Read Serial Prog. Fuse	H	H	12V	H	H	L	H	@PA0	X

Notes:

1. Chip Erase and Serial Programming Fuse requires a 10 ms $\overline{PROG}$ pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than $FF.
2. PD4 is pulled Low during programming to indicate RDY/$\overline{BSY}$.
3. 'X' = Don't care.

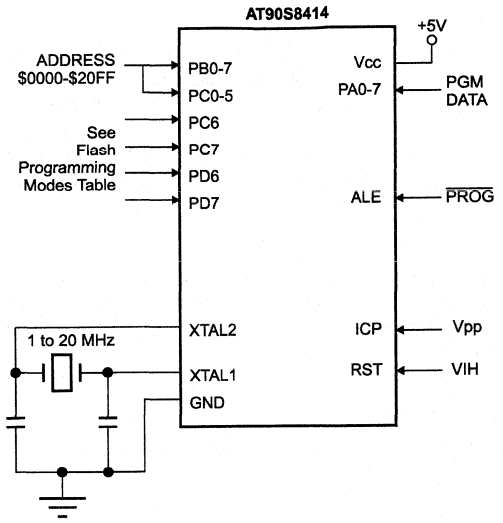


Figure 53: Parallel Programming

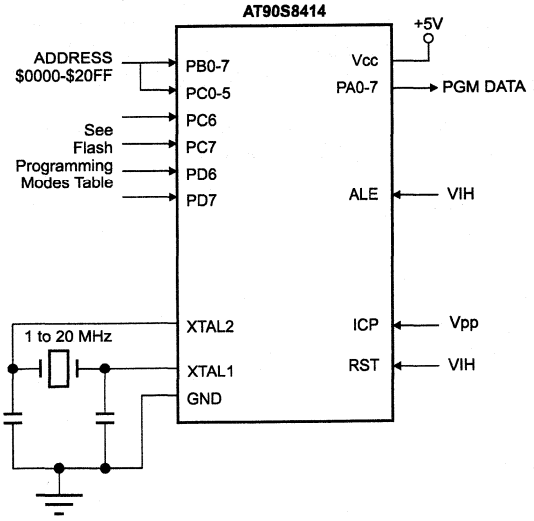


Figure 54: Parallel Verify

Serial Downloading

Both the Program and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to VCC. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before program/erase operations can be executed.

An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction. The Chip Erase operation turns the content of every memory location in both the Program and Data arrays into $FF.

The Program and Data memory arrays have separate address spaces:
$0000 to $1FFF for Program memory and $0000 to $00FF for Data memory.
Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 20 MHz oscillator clock, the maximum SCK frequency is 500 kHz.

SERIAL PROGRAMMING ALGORITHM

To program and verify the AT90S8414 in the serial programming mode, the following sequence is recommended (See three byte instruction formats in Table 23):

1. *Power-up sequence:*
   Apply power between VCC and GND.
   Set RST pin to 'H'.
   If a crystal is not connected across pins XTAL1 and XTAL2, apply a 1 MHz to 20 MHz clock to the XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/PB5. The frequency of the shift clock supplied at pin SCK/PB7 needs to be less than the CPU clock at XTAL1 divided by 40.
3. The Program or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/PB6.
   $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle, which typically takes less than 2.5 ms.
5. At the end of the programming session, RST can be set low to commence normal operation.
6. *Power-off sequence (if needed):*
   Set XTAL1 to 'L' (if a crystal is not used).
   Set RST to 'L'
   Turn Vcc power off.

Table 23: Serial Programming Instruction Set

Instruction	Instruction Format Byte 1	Byte 2	Byte3	Operation
Programming Enable	1010 1100	0101 0011	xxxx xxxx	Enable Serial Programming after RST goes high.
Chip Erase	1010 1100	xxxx x100	xxxx xxxx	Chip erase both 8K & 256 byte memory arrays
Read Program Memory	aaaa a001	bbbb bbbb	oooo oooo	Read data **o** from Program memory at address **a:b**
Write Program Memory	aaaa a010	bbbb bbbb	iiii iiii	Write data **i** to Program memory at address **a:b**
Read Data Memory	0000 0101	bbbb bbbb	oooo oooo	Read data **o** from Data memory at address **b**
Write Data Memory	0000 0110	bbbb bbbb	iiii iiii	Write data **i** to Data memory at address **b**
Write Lock Bits	1010 1100	0*12*x x111	xxxx xxxx	Write lock bits. Set bits ***1,2***='0' to program lock bits.

Note: **a** = address high bits
b = address low bits
o = data out
i = data in
x = don't care
1 = lock bit 1
2 = lock bit 2

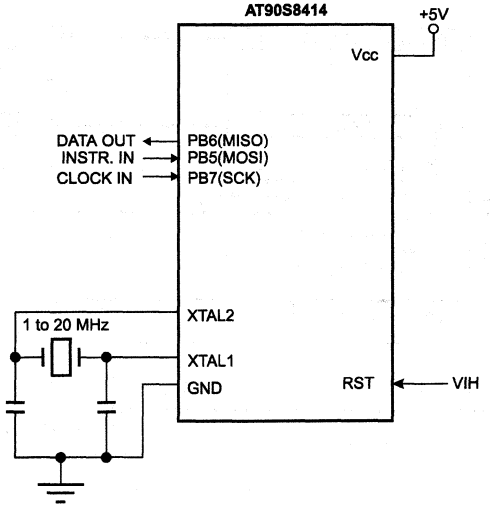


Figure 55: Serial Programming and Verify

When *writing* serial data to the AT90S8414, data is clocked on the *rising* edge of CLK.
When *reading* data from the AT90S8414, data is clocked on the *falling* edge of CLK. See Figure 57 for an explanation.

Programming Characteristics

Table 24: Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, Vcc = 5.0V±10%.

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		1.0	mA
$1/t_{CK}$	Oscillator Clock Frequency	1	20	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	48 t_{CK}		
t_{GHAX}	Address Setup After $\overline{PROG}$	48 t_{CK}		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	48 t_{CK}		
t_{GHDX}	Data Hold after $\overline{PROG}$	48 t_{CK}		
t_{EHSH}	PC7 ($\overline{ENABLE}$) High to V_{PP}	48 t_{CK}		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48 t_{CK}	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid	0	48 t_{CK}	
t_{EHQV}	Data Float after $\overline{ENABLE}$		48 t_{CK}	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

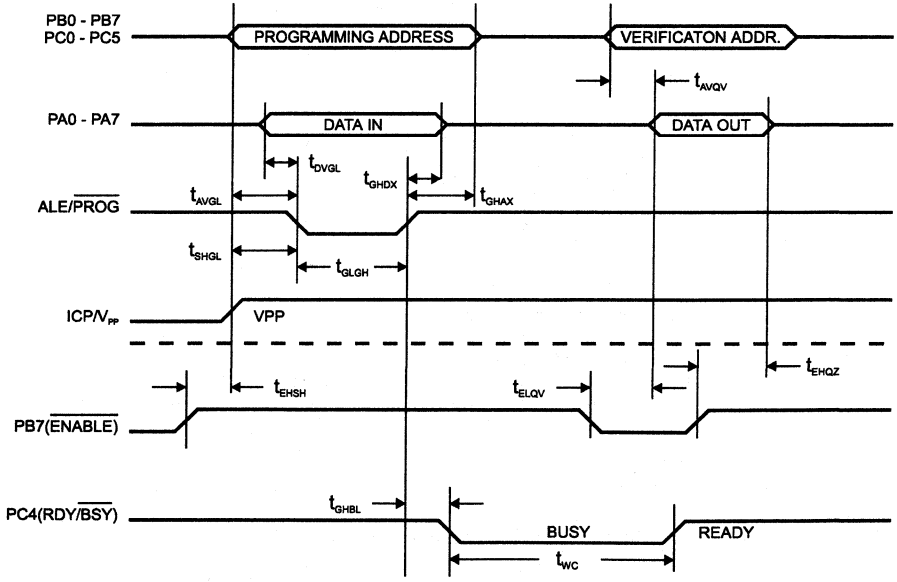


Figure 56: Flash/EEPROM Programming and Verfication Waveforms - Parallel Mode

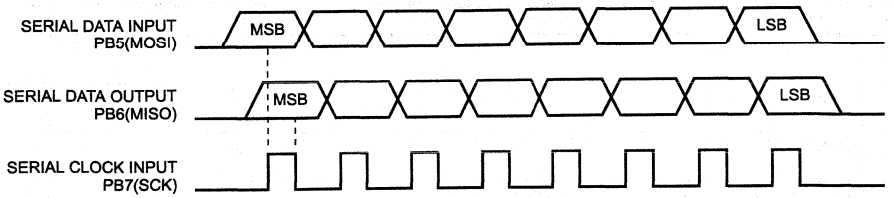


Figure 57: Serial Downloading Waveforms

4

Absolute Maximum Ratings

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin with respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	25.0mA

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

D.C. Characteristics

T_A=-40°C to 85°C Vcc=2.7V to 6.0V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IL}	Input Low Voltage		-0.5		0.2 Vcc - 0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RESET)	0.2 VCC + 0.9		VCC + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RESET)	0,7 VCC		VCC + 0.5	V
V_{OL}	Output Low Voltage[1] (Ports B,C,D)	I_{OL} = 20 mA, VCC = 5 V I_{OL} = 10 mA, VCC = 2.7 V			0.5	V
V_{OH}	Output High Voltage (Ports B,C,D)	I_{HI} = 10 mA, VCC = 5 V I_{HI} = 5 mA, VCC = 2.7 V	4.5			V
I_{OH}	Output Source Current (Ports B,C,D)	VCC = 5 V VCC = 2.7 V			10 5	mA
I_{OL}	Output Sink Current (Port B,C,D)	VCC = 5 V VCC = 2.7 V			20 10	mA
RRST	Reset Pulldown Resistor		10		50	KΩ
I_{CC}	Power Supply Current	Active Mode, 3V, 4MHz		3.5		mA
		Idle Mode 3V, 4MHz		1000		µA
I_{CC}	Power Down Mode[2]	WDT enabled, 3V		50		µA
		WDT disabled, 3V		<1		µA

Notes:

1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10mA
   Maximum total I_{OL} for all output pins: 80mA
   Port A: 26 mA
   Ports A, B, D 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

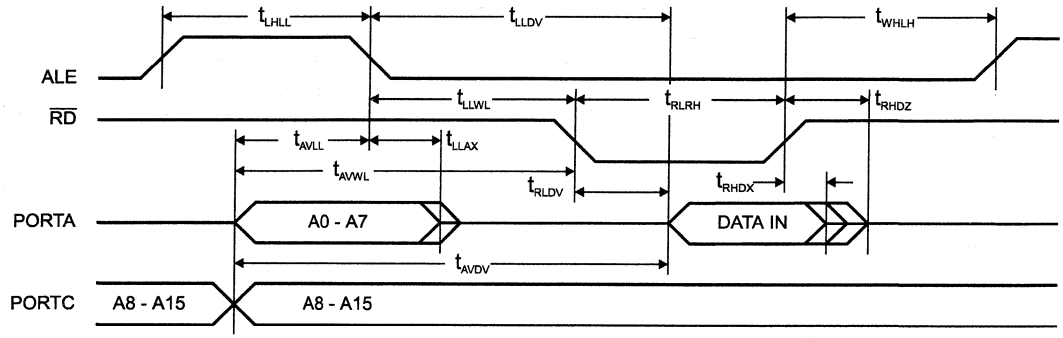
A.C. Characteristics

Load capacitance for Port A & ALE = 100 pF; Load capacitance for all outputs = 80 pF.

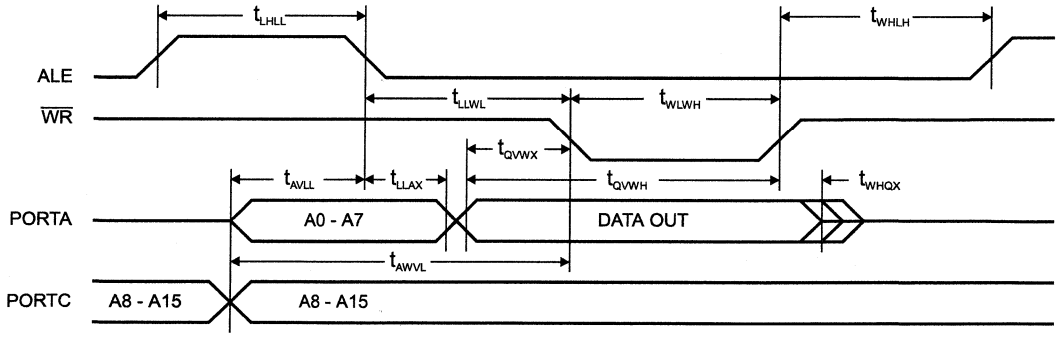
Symbol	Parameter	10 MHz Oscillator Min	10 MHz Oscillator Max	Variable Oscillator Min	Variable Oscillator Max	Units
$1/t_{CK}$	Oscillator Frequency			0	20	MHz
t_{LHLL}	ALE Pulse Width	50		$t_{ck}/2$		ns
t_{AVLL}	Address Valid to ALE Low	40		$t_{ck}/2$-10		ns
t_{LLAX}	Address Hold After ALE Low	10		10		ns
t_{RLRH}	$\overline{RD}$ Pulse Width	150		$3\,t_{ck}/2$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	50		$t_{ck}/2$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		10		10	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		20		20	ns
t_{LLDV}	ALE Low to Valid Data In		60		$t_{ck}/2$+10	ns
t_{AVDV}	Address to Valid Data In		100		t_{ck}	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$		100		t_{ck}	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	90		t_{ck}-10		ns
t_{QVWX}	Data Valid to $\overline{WR}$ High	60		$t_{ck}/2$+10		ns
t_{WHQX}	Data Hold after $\overline{WR}$	15		15		ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High		100		t_{ck}	ns

4

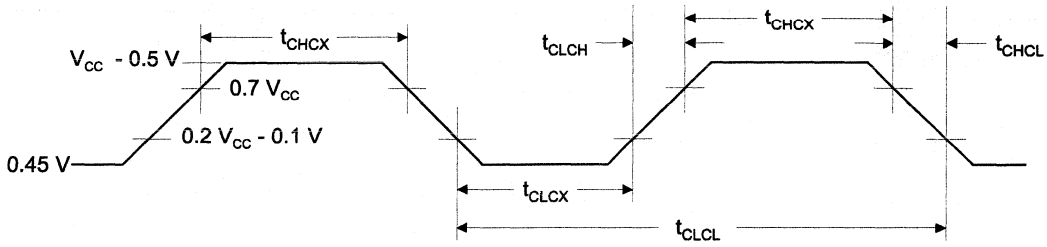
External Data Memory Read Cycle



External Memory Write Cycle



External Clock Drive Waveforms



4

External Clock Drive

Symbol	Parameter	VCC = 2.7 V to 6.0 V		VCC = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	10	0	24	MHz
t_{CLCL}	Clock Period	100		41.7		ns
t_{CHCX}	High Time	40		16.7		ns
t_{CLCX}	Low Time	40		16.7		ns
t_{CLCH}	Rise Time		10		4.15	ns
t_{CHCL}	Fall Time		10		4.15	ns

Ordering Information

Ordering Code	Package	Operation Range
AT90S8414-JC AT90S8414-PC	44J 40P6	Commercial (0°C to 70°C)
AT90S8414-JI AT90S8414-PI	44J 40P6	Industrial (-40°C to 85°C)

Package Type	
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual in Line Package (PDIP)

AT90S8414 Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
$3F	SREG	I	T	O	S	V	N	Z	C	4-23
$3E	SPH	-	-	-	-	-	-	-	SP8	4-24
$3D	SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	4-24
$3C	Reserved									
$3B	GIMSK	INT1	INT0	-	-	-	-	-	-	4-25
$3A	Reserved									
$39	TIMSK	TOIE1	OCIE1A	OCIE1B	-	TICIE1	-	TOIE0	OCIE0	4-26
$38	TIFR	TOV1	OCF1A	OCF1B	-	ICF1	-	TOV0	OCF0	4-27
$37	Reserved									
$36	Reserved									
$35	MCUCR	SRE	SRW	SE	SM	ISC11	ISC10	ISC01	ISC00	4-28
$34	Reserved									
$33	TCCR0	-	-	COM01	COM00	CTC0	CS02	CS01	CS00	4-31
$32	TCNT0	Timer/Counter0 (8 Bit)								4-32
$31	OCR0	Timer/Counter0 Output Compare Register								4-32
$30	Reserved									
$2F	TCCR1A	COM1A1	COM1A0	COM1B1	COM1B0	-	-	-	PWM1	4-35
$2E	TCCR1B	ICNC1	ICES1	-	-	CTC1	CS12	CS11	CS10	4-35
$2D	TCNT1H	Timer/Counter1 - Counter Register High Byte								4-36
$2C	TCNT1L	Timer/Counter1 - Counter Register Low Byte								4-36
$2B	OCR1AH	Timer/Counter1 - Output Compare Register A High Byte								4-37
$2A	OCR1AL	Timer/Counter1 - Output Compare Register A Low Byte								4-37
$29	OCR1BH	Timer/Counter1 - Output Compare Register B High Byte								4-37
$28	OCR1BL	Timer/Counter1 - Output Compare Register B Low Byte								4-37
$27	Reserved									
$26	Reserved									
$25	ICR1H	Timer/Counter1 - Input Capture Register High Byte								4-37
$24	ICR1L	Timer/Counter1 - Input Capture Register Low Byte								4-37
$23	Reserved									
$22	Reserved									
$21	WDTCR	-	-	-	-	WDE	WDP2	WDP1	WDP0	4-39
$20	Reserved									
$1F	Reserved									
$1E	EEAR	EEPROM Address Register								4-40
$1D	EEDR	EEPROM Data Register								4-40
$1C	EECR	-	-	-	-	-	-	EEWE	EERE	4-41
$1B	PORTA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	4-54
$1A	DDRA	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	4-54
$19	PINA	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	4-55
$18	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	4-57
$17	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	4-57
$16	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	4-57
$15	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0	4-62
$14	DDRC	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	4-62
$13	PINC	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	4-62
$12	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0	4-64
$11	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	4-64
$10	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	4-64
$0F	SPDR	SPI Data Register								4-45
$0E	SPSR	SPIF	WCOL	-	-	-	-	-	-	4-45
$0D	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	4-44
$0C	UDR	UART I/O Data Register								4-49
$0B	USR	RXC	TXC	UDRE	FE	OR	-	-	-	4-49
$0A	UCR	RXCIE	TXCIE	UDRIE	RXEN	TXEN	CHR9	RXB8	TXB8	4-50
$09	UBRR	UART Baud Rate Register								4-52
$08	ACSR	-	-	ACO	ACI	ACIE	ACIC	ACIS1	ACIS0	4-53
...	Reserved									
$00	Reserved									

AT90S8414 Instruction Set Summary

Mnemonic	Operands	Description	Operation	Flags	#Clocks
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add two Registers	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry two Registers	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract two Registers	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Constant from Register	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry two Registers	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract with Carry Constant from Reg.	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND Registers	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND Register and Constant	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR Registers	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR Register and Constant	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR Registers	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FF - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
RCALL	k	Relative Subroutine Call	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2
CP	Rd,Rr	Compare	Rd - Rr	Z, N,V,C,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z, N,V,C,H	1
CPI	Rd,K	Compare Register with Immediate	Rd - K	Z, N,V,C,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register is Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than Zero, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

(continued)

AT90S8414 Instruction Set Summary (Continued)

Mnemonic	Operands	Description	Operation	Flags	#Clocks
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2
BIT AND BIT-TEST INSTRUCTIONS					
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep function)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR/timer)	None	1

AVR Instruction Set

This section describes all instructions for the 8-bit AVR in detail. For a specific device please refer to the specific Instruction Set Summary in the hardware description.

Addressing modes are described in detail in the hardware description for each device.

Instruction Set Nomenclature:

Status Register (SREG):
SREG: Status register
C: Carry flag in status register
Z: Zero flag in status register
N: Negative flag in status register
V: Twos complement overflow indicator
S: N ⊕ V, For signed tests
H: Half Carry flag in the status register
T: Transfer bit used by BLD and BST instructions
I: Global interrupt enable/disable flag

Registers and operands:
Rd: Destination (and source) register in the register file
Rr: Source register in the register file
R: Result after instruction is executed
K: Constant literal or byte data (8 bit)
k: Constant address data for program counter
b: Bit in the register file (3 bit)
s: Bit in the status register (3 bit)

X,Y,Z: Indirect address register (X=R27:R26, Y=R29:R28 and Z=R31:R30)
P: I/O port address
q: Displacement for direct addressing (6 bit)

I/O Registers
RAMPX, RAMPY, RAMPZ: Registers concatenated with the X, Y and Z registers enabling indirect addressing of the whole SRAM area on MCUs with more than 64K bytes SRAM.

Stack:
STACK: Stack for return address and pushed registers
SP: Stack Pointer to STACK

Opcode:
X: Don't care

Flags:
⇔: Flag affected by instruction
0: Flag cleared by instruction
1: Flag set by instruction
-: Flag not affected by instruction

Conditional Branch Summary

Test	Boolean	Mnemonic	Complementary	Boolean	Mnemonic	Comment
Rd > Rr	Z•(N ⊕ V) = 0	BRLT*	Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE*	Signed
Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Rd < Rr	(N ⊕ V) = 1	BRLT	Signed
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Signed
Rd ≤ Rr	Z+(N ⊕ V) = 1	BRGE*	Rd > Rr	Z•(N ⊕ V) = 0	BRLT*	Signed
Rd < Rr	(N ⊕ V) = 1	BRLT	Rd ≥ Rr	(N ⊕ V) = 0	BRGE	Signed
Rd > Rr	C + Z = 0	BRLO*	Rd ≤ Rr	C + Z = 1	BRSH*	Unsigned
Rd ≥ Rr	C = 0	BRSH/BRCC	Rd < Rr	C = 1	BRLO/BRCS	Unsigned
Rd = Rr	Z = 1	BREQ	Rd ≠ Rr	Z = 0	BRNE	Unsigned
Rd ≤ Rr	C + Z = 1	BRSH*	Rd > Rr	C + Z = 0	BRLO*	Unsigned
Rd < Rr	C = 1	BRLO/BRCS	Rd ≥ Rr	C = 0	BRSH/BRCC	Unsigned
Carry	C = 1	BRCS	No carry	C = 0	BRCC	Simple
Negative	N = 1	BRMI	Positive	N = 0	BRPL	Simple
Overflow	V = 1	BRVS	No overflow	V = 0	BRVC	Simple
Zero	Z = 1	BREQ	Not zero	Z = 0	BRNE	Simple

* Interchange Rd and Rr in the operation before the test. i.e. CP Rd,Rr → CP Rr,Rd

Complete Instruction Set Summary

Mnemonics	Operands	Description	Operation	Flags	#Clock Note
ARITHMETIC AND LOGIC INSTRUCTIONS					
ADD	Rd, Rr	Add without Carry	Rd ← Rd + Rr	Z,C,N,V,H	1
ADC	Rd, Rr	Add with Carry	Rd ← Rd + Rr + C	Z,C,N,V,H	1
SUB	Rd, Rr	Subtract without Carry	Rd ← Rd - Rr	Z,C,N,V,H	1
SUBI	Rd, K	Subtract Immediate	Rd ← Rd - K	Z,C,N,V,H	1
SBC	Rd, Rr	Subtract with Carry	Rd ← Rd - Rr - C	Z,C,N,V,H	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd ← Rd - K - C	Z,C,N,V,H	1
AND	Rd, Rr	Logical AND	Rd ← Rd • Rr	Z,N,V	1
ANDI	Rd, K	Logical AND with Immediate	Rd ← Rd • K	Z,N,V	1
OR	Rd, Rr	Logical OR	Rd ← Rd v Rr	Z,N,V	1
ORI	Rd, K	Logical OR with Immediate	Rd ← Rd v K	Z,N,V	1
EOR	Rd, Rr	Exclusive OR	Rd ← Rd ⊕ Rr	Z,N,V	1
COM	Rd	One's Complement	Rd ← $FF - Rd	Z,C,N,V	1
NEG	Rd	Two's Complement	Rd ← $00 - Rd	Z,C,N,V,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd ← Rd v K	Z,N,V	1
CBR	Rd,K	Clear Bit(s) in Register	Rd ← Rd • ($FFh - K)	Z,N,V	1
INC	Rd	Increment	Rd ← Rd + 1	Z,N,V	1
DEC	Rd	Decrement	Rd ← Rd - 1	Z,N,V	1
TST	Rd	Test for Zero or Minus	Rd ← Rd • Rd	Z,N,V	1
CLR	Rd	Clear Register	Rd ← Rd ⊕ Rd	Z,N,V	1
SER	Rd	Set Register	Rd ← $FF	None	1
MUL	Rd,Rr	Multiply Unsigned	R1, R0 ← Rd × Rr	C	2 √

√) Not available in base-line microcontrollers

(continued)

Complete Instruction Set Summary (continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock Note
BRANCH INSTRUCTIONS					
RJMP	k	Relative Jump	PC ← PC + k + 1	None	2
IJMP		Indirect Jump to (Z)	PC ← Z	None	2
JMP	k	Jump	PC ← k	None	3
RCALL	k	Relative Call Subroutine	PC ← PC + k + 1	None	3
ICALL		Indirect Call to (Z)	PC ← Z	None	3
CALL	k	Call Subroutine	PC ← k	None	4
RET		Subroutine Return	PC ← STACK	None	4
RETI		Interrupt Return	PC ← STACK	I	4
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC ← PC + 2 or 3	None	1 / 2
CP	Rd,Rr	Compare	Rd - Rr	Z,C,N,V,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C	Z,C,N,V,H	1
CPI	Rd,K	Compare with Immediate	Rd - K	Z,C,N,V,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b)=0) PC ← PC + 2 or 3	None	1 / 2
SBRS	Rr, b	Skip if Bit in Register Set	if (Rr(b)=1) PC ← PC + 2 or 3	None	1 / 2
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC←PC+k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC←PC+k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC ← PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC ← PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC ← PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC ← PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC ← PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC ← PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC ← PC+ k + 1	None	1 / 2
BRLT	k	Branch if Less Than, Signed	if (N ⊕ V= 1) then PC ← PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC ← PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC ← PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC ← PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC ← PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC ← PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1 / 2

(continued)

Complete Instruction Set Summary (continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock Note
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Copy Register	Rd ← Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Increment	Rd ← (X), X ← X + 1	None	2
LD	Rd, -X	Load Indirect and Pre-Decrement	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Increment	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, -Y	Load Indirect and Pre-Decrement	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Increment	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Decrement	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Increment	(X) ← Rr, X ← X + 1	None	2
ST	-X, Rr	Store Indirect and Pre-Decrement	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Increment	(Y) ← Rr, Y ← Y + 1	None	2
ST	-Y, Rr	Store Indirect and Pre-Decrement	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Increment	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Decrement	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1
PUSH	Rr	Push Register on Stack	STACK ← Rr	None	2
POP	Rd	Pop Register from Stack	Rd ← STACK	None	2

(continued)

Complete Instruction Set Summary (continued)

Mnemonics	Operands	Description	Operation	Flags	#Clock Note
BIT AND BIT-TEST INSTRUCTIONS					
LSL	Rd	Logical Shift Left	Rd(n+1)←Rd(n),Rd(0)←0,C←Rd(7)	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n)←Rd(n+1),Rd(7)←0,C←Rd(0)	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)←Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)←Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0) ↔ Rd(7..4)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Two's Complement Overflow	V ← 1	V	1
CLV		Clear Two's Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR)	None	1

ADC - Add with Carry

Description:
Adds two registers and the contents of the C flag and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd + Rr + C

	Syntax:	Operands:	Program Counter:
(i)	ADC Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0001	11rd	dddd	rrrr

Status Register (SREG) Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $Rd3 \cdot Rr3 + Rr3 \cdot \overline{R3} + \overline{R3} \cdot Rd3$
Set if there was a carry from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot Rr7 \cdot \overline{R7} + \overline{Rd7} \cdot \overline{Rr7} \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $Rd7 \cdot Rr7 + Rr7 \cdot \overline{R7} + \overline{R7} \cdot Rd7$
Set if there was carry from the MSB of the result; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
                ; Add R1:R0 to R3:R2
add   r2,r0 ; Add low byte
adc   r3,r1 ; Add with carry high byte
```

Words: 1 (2 bytes)
Cycles: 1

ADD - Add without Carry

Description:
Adds two registers without the C flag and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd + Rr

	Syntax:	Operands:	Program Counter:
(i)	ADD Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0000	11rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $Rd3 \cdot Rr3 + Rr3 \cdot \overline{R3} + \overline{R3} \cdot Rd3$
Set if there was a carry from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot Rr7 \cdot \overline{R7} + \overline{Rd7} \cdot \overline{Rr7} \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $Rd7 \cdot Rr7 + Rr7 \cdot \overline{R7} + \overline{R7} \cdot Rd7$
Set if there was carry from the MSB of the result; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
add   r1,r2        ; Add r2 to r1 (r1=r1+r2)
add   r28,r28      ; Add r28 to itself (r28=r28+r28)
```

Words: 1 (2 bytes)
Cycles: 1

AND - Logical AND

Description:
Performs the logical AND between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd • Rr

	Syntax:	Operands:	Program Counter:
(i)	AND Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0010	00rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
and   r2,r3       ; Bitwise and r2 and r3, result in r2
ldi   r16,1       ; Set bitmask 0000 0001 in r16
and   r2,r16      ; Isolate bit 0 in r2
```

Words: 1 (2 bytes)
Cycles: 1

ANDI - Logical AND with Immediate

Description:
Performs the logical AND between the contents of register Rd and a constant and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd • K

	Syntax:	Operands:	Program Counter:
(i)	ANDI Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC ← PC + 1

16 bit Opcode:

0111	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
andi   r17,$0F      ; Clear upper nibble of r17
andi   r18,$10      ; Isolate bit 4 in r18
andi   r19,$AA      ; Clear odd bits of r19
```

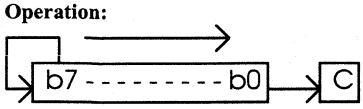
Words: 1 (2 bytes)
Cycles: 1

ASR - Arithmetic Shift Right

Description:
Shifts all bits in Rd one place to the right. Bit 7 is held constant. Bit 0 is loaded into the C flag of the SREG. This operation effectively divides a twos complement value by two without changing its sign. The carry flag can be used to round the result.

Operation:

(i)



	Syntax:	Operands:	Program Counter:
(i)	ASR Rd	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0101

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S: $N \oplus V$, For signed tests.

V: $N \oplus C$ (For N and C after the shift)
Set if (N is set and C is clear) or (N is clear and C is set); Cleared otherwise (for values of N and C after the shift).

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: Rd0
Set if, before the shift, the LSB of Rd was set; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
ldi   r16,$10      ; Load decimal 16 into r16
asr   r16          ; r16=r16 / 2
ldi   r17,$FC      ; Load -4 in r17
asr   r17          ; r17=r17/2
```

Words: 1 (2 bytes)
Cycles: 1

BCLR - Bit Clear in SREG

Description:
Clears a single flag in SREG.

	Operation:		
(i)	SREG(s) ← 0		
	Syntax:	**Operands:**	**Program Counter:**
(i)	BCLR s	0 ≤ s ≤ 7	PC ← PC + 1

16 bit Opcode:

1001	0100	1sss	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔

I: 0 if s = 7; Unchanged otherwise.

T: 0 if s = 6; Unchanged otherwise.

H: 0 if s = 5; Unchanged otherwise.

S: 0 if s = 4; Unchanged otherwise.

V: 0 if s = 3; Unchanged otherwise.

N: 0 if s = 2; Unchanged otherwise.

Z: 0 if s = 1; Unchanged otherwise.

C: 0 if s = 0; Unchanged otherwise.

Example:

```
bclr  0          ; Clear carry flag
bclr  7          ; Disable interrupts
```

Words: 1 (2 bytes)
Cycles: 1

BLD - Bit Load from the T Flag in SREG to a Bit in Register.

Description:
Copies the T flag in the SREG (status register) to bit b in register Rd.

Operation:
(i) Rd(b) ← T

	Syntax:	Operands:	Program Counter:
(i)	BLD Rd,b	0 ≤ d ≤ 31, 0 ≤ b ≤ 7	PC ← PC + 1

16 bit Opcode:

1111	100d	dddd	Xbbb

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
                        ; Copy bit
bst   r1,2              ; Store bit 2 of r1 in T flag
bld   r0,4              ; Load T flag into bit 4 of r0
```

Words: 1 (2 bytes)
Cycles: 1

BRBC - Branch if Bit in SREG is Cleared

Description:
Conditional relative branch. Tests a single bit in SREG and branches relatively to PC if the bit is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form.

Operation:
(i) If SREG(s) = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRBC s,k	0 ≤ s ≤ 7, -64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	ksss

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            cpi   r20,5       ; Compare r20 to the value 5
            brbc  1,noteq     ; Branch if zero flag cleared
            ...
noteq:      nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRBS - Branch if Bit in SREG is Set

Description:
Conditional relative branch. Tests a single bit in SREG and branches relatively to PC if the bit is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form.

Operation:
(i) If SREG(s) = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRBS s,k	$0 \le s \le 7$, $-64 \le k \le +63$	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	ksss

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            bst   r0,3          ; Load T bit with bit 3 of r0
            brbs  6,bitset      ; Branch T bit was set
            ...
bitset:     nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRCC - Branch if Carry Cleared

Description:
Conditional relative branch. Tests the Carry flag (C) and branches relatively to PC if C is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 0,k).

Operation:
(i) If C = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRCC k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            add   r22,r23      ; Add r23 to r22
            brcc  nocarry      ; Branch if carry cleared
            ...
nocarry:    nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRCS - Branch if Carry Set

Description:
Conditional relative branch. Tests the Carry flag (C) and branches relatively to PC if C is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 0,k).

Operation:
(i) If C = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRCS k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        cpi   r26,$56     ; Compare r26 with $56
        brcs  carry       ; Branch if carry set
        ...
carry:  nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BREQ - Branch if Equal

Description:
Conditional relative branch. Tests the Zero flag (Z) and branches relatively to PC if Z is set. If the instruction is executed immediately after any of the instructions CP, CPI, SUB or SUBI, the branch will occur if and only if the unsigned or signed binary number represented in Rd was equal to the unsigned or signed binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 1,k).

Operation:
(i) If Rd = Rr (Z = 1) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BREQ k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k001

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        cp    r1,r0       ; Compare registers r1 and r0
        breq  equal       ; Branch if registers equal
        ...
equal:  nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRGE - Branch if Greater or Equal (Signed)

Description:
Conditional relative branch. Tests the Signed flag (S) and branches relatively to PC if S is cleared. If the instruction is executed immediately after any of the instructions CP, CPI, SUB or SUBI, the branch will occur if and only if the signed binary number represented in Rd was greater than or equal to the signed binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 4,k).

Operation:
(i) If Rd ≥ Rr (N ⊕ V = 0) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRGE k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k100

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            cp    r11,r12     ; Compare registers r11 and r12
            brge  greateq     ; Branch if r11 >= r12 (signed)
            ...
greateq:    nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRSH - Branch if Same or Higher (Unsigned)

Description:
Conditional relative branch. Tests the Carry flag (C) and branches relatively to PC if C is cleared. If the instruction is executed immediately after execution of any of the instructions CP, CPI, SUB or SUBI the branch will occur if and only if the unsigned binary number represented in Rd was greater than or equal to the unsigned binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 0,k).

Operation:
(i) If Rd ≥Rr (C = 0) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRSH k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            subi   r19,4        ; Subtract 4 from r19
            brsh   highsm       ; Branch if r19 >= 4 (unsigned)
            ...
highsm:     nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRID - Branch if Global Interrupt is Disabled

Description:
Conditional relative branch. Tests the Global Interrupt flag (I) and branches relatively to PC if I is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 7,k).

Operation:
(i) If I = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRID k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        brid intdis       ; Branch if interrupt disabled
        ...
intdis: nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRIE - Branch if Global Interrupt is Enabled

Description:
Conditional relative branch. Tests the Global Interrupt flag (I) and branches relatively to PC if I is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 7,k).

Operation:
(i) If I = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRIE k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        brie  inten       ; Branch if interrupt enabled
        ...
inten:  nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRLO - Branch if Lower (Unsigned)

Description:
Conditional relative branch. Tests the Carry flag (C) and branches relatively to PC if C is set. If the instruction is executed immediately after any of the instructions CP, CPI, SUB or SUBI, the branch will occur if and only if the unsigned binary number represented in Rd was smaller than the unsigned binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 0,k).

Operation:
(i) If Rd < Rr (C = 1) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRLO k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            eor   r19,r19     ; Clear r19
loop:       inc   r19         ; Increase r19
            ...
            cpi   r19,$10     ; Compare r19 with $10
            brlo  loop        ; Branch if r19 < $10 (unsigned)
            nop               ; Exit from loop (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRLT - Branch if Less Than (Signed)

Description:
Conditional relative branch. Tests the Signed flag (S) and branches relatively to PC if S is set. If the instruction is executed immediately after any of the instructions CP, CPI, SUB or SUBI, the branch will occur if and only if the signed binary number represented in Rd was less than the signed binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 4,k).

Operation:
(i) If Rd < Rr (N ⊕ V = 1) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRLT k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k100

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            cp    r16,r1      ; Compare r16 to r1
            brlt  less        ; Branch if r16 < r1 (signed)
            ...
less:       nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRMI - Branch if Minus

Description:
Conditional relative branch. Tests the Negative flag (N) and branches relatively to PC if N is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 2,k).

Operation:
(i) If N = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRMI k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k010

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            subi   r18,4        ; Subtract 4 from r18
            brmi   negative     ; Branch if result negative
            ...
negative:   nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRNE - Branch if Not Equal

Description:
Conditional relative branch. Tests the Zero flag (Z) and branches relatively to PC if Z is cleared. If the instruction is executed immediately after any of the instructions CP, CPI, SUB or SUBI, the branch will occur if and only if the unsigned or signed binary number represented in Rd was not equal to the unsigned or signed binary number represented in Rr. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 1,k).

Operation:
(i) If Rd ≠ Rr (Z = 0) then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRNE k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k001

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            eor   r27,r27     ; Clear r27
loop:       inc   r27         ; Increase r27
            ...
            cpi   r27,5       ; Compare r27 to 5
            brne  loop        ; Branch if r27<>5
            nop               ; Loop exit (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRHC - Branch if Half Carry Flag is Cleared

Description:
Conditional relative branch. Tests the Half Carry flag (H) and branches relatively to PC if H is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 5,k).

Operation:
(i) If H = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRHC k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k101

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        brhc oclear       ; Branch if half carry flag cleared
        ...
oclear: nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRHS - Branch if Half Carry Flag is Set

Description:
Conditional relative branch. Tests the Half Carry flag (H) and branches relatively to PC if H is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 5,k).

Operation:
(i) If H = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRHS k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k101

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:
```
        brhs  oset        ; Branch if half carry flag set
        ...
oset:   nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRPL - Branch if Plus

Description:
Conditional relative branch. Tests the Negative flag (N) and branches relatively to PC if N is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 2,k).

Operation:
(i) If N = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRPL k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k010

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            subi  r26,$50      ; Subtract $50 from r26
            brpl  positive     ; Branch if r26 positive
            ...
positive:   nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRTC - Branch if the T Flag is Cleared

Description:
Conditional relative branch. Tests the T flag and branches relatively to PC if T is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 6,k).

Operation:
(i) If T = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRTC k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k110

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            bst   r3,5        ; Store bit 5 of r3 in T flag
            brtc  tclear      ; Branch if this bit was cleared
            ...
tclear:     nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRTS - Branch if the T Flag is Set

Description:
Conditional relative branch. Tests the T flag and branches relatively to PC if T is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 6,k).

Operation:
(i) If T = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRTS k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k110

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            bst   r3,5        ; Store bit 5 of r3 in T flag
            brts  tset        ; Branch if this bit was set
            ...
tset:       nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRVC - Branch if Overflow Cleared

Description:
Conditional relative branch. Tests the Overflow flag (V) and branches relatively to PC if V is cleared. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBC 3,k).

Operation:
(i) If V = 0 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRVC k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	01kk	kkkk	k011

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
        add   r3,r4       ; Add r4 to r3
        brvc  noover      ; Branch if no overflow
        ...
noover: nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BRVS - Branch if Overflow Set

Description:
Conditional relative branch. Tests the Overflow flag (V) and branches relatively to PC if V is set. This instruction branches relatively to PC in either direction (PC-64≤destination≤PC+63). The parameter k is the offset from PC and is represented in two's complement form. (Equivalent to instruction BRBS 3,k).

Operation:
(i) If V = 1 then PC ← PC + k + 1, else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	BRVS k	-64 ≤ k ≤ +63	PC ← PC + k + 1 PC ← PC + 1, if condition is false

16 bit Opcode:

1111	00kk	kkkk	k011

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            add   r3,r4         ; Add r4 to r3
            brvs  overfl        ; Branch if overflow
            ...
overfl:     nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false
2 if condition is true

BSET - Bit Set in SREG

Description:
Sets a single flag or bit in SREG.

Operation:
(i) SREG(s) ← 1

	Syntax:	Operands:	Program Counter:
(i)	BSET s	0 ≤ s ≤ 7	PC ← PC + 1

16 bit Opcode:

1001	0100	0sss	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
⇔	⇔	⇔	⇔	⇔	⇔	⇔	⇔

I: 1 if s = 7; Unchanged otherwise.

T: 1 if s = 6; Unchanged otherwise.

H: 1 if s = 5; Unchanged otherwise.

S: 1 if s = 4; Unchanged otherwise.

V: 1 if s = 3; Unchanged otherwise.

N: 1 if s = 2; Unchanged otherwise.

Z: 1 if s = 1; Unchanged otherwise.

C: 1 if s = 0; Unchanged otherwise.

Example:

```
bset  6      ; Set T flag
bset  7      ; Enable interrupt
```

Words: 1 (2 bytes)
Cycles: 1

BST - Bit Store from Bit in Register to T Flag in SREG

Description:
Stores bit b from Rd to the T flag in SREG (status register).

Operation:
(i) T ← Rd(b)

	Syntax:	Operands:	Program Counter:
(i)	BST Rd,b	$0 \le d \le 31, 0 \le b \le 7$	PC ← PC + 1

16 bit Opcode:

1111	101d	dddd	Xbbb

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	⇔	-	-	-	-	-	-

T: 0 if bit b in Rd is cleared. Set to 1 otherwise.

Example:

```
                        ; Copy bit
        bst   r1,2      ; Store bit 2 of r1 in T flag
        bld   r0,4      ; Load T into bit 4 of r0
```

Words: 1 (2 bytes)
Cycles: 1

CALL - Long Call to a Subroutine

Description:
Calls to a subroutine within the entire program memory. The return address (to the instruction after the CALL) will be stored onto the stack. (See also RCALL).

Operation:

(i) PC ← k Devices with 16 bits PC, 128K bytes program memory maximum.
(ii) PC ← k Devices with 22 bits PC, 8M bytes program memory maximum.

	Syntax:	Operands:	Program Counter:	Stack
(i)	CALL k	0 ≤ k ≤ 64K	PC ← k	STACK ← PC+2 SP ← SP-2, (2 bytes, 16 bits)
(ii)	CALL k	0 ≤ k ≤ 4M	PC ← k	STACK ← PC+2 SP ← SP-3 (3 bytes, 22 bits)

32 bit Opcode:

1001 kkkk	010k kkkk	kkkk kkkk	111k kkkk

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            mov   r16,r0      ; Copy r0 to r16
            call  check       ; Call subroutine
            nop               ; Continue (do nothing)
            ...
check:      cpi   r16,$42     ; Check if r16 has a special value
            breq  error       ; Branch if equal
            ret               ; Return from subroutine
            ...
error:      rjmp  error       ; Infinite loop
```

Words: 2 (4 bytes)
Cycles: 4

CBR - Clear Bits in Register

Description:
Clears the specified bits in register Rd. Performs the logical AND between the contents of register Rd and the complement of the constant mask K. The result will be placed in register Rd.

	Operation:		
(i)	Rd ← Rd • ($FF - K)		
	Syntax:	**Operands:**	**Program Counter:**
(i)	CBR Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC ← PC + 1

16 bit Opcode: See ANDI with K complemented.

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: $N \oplus V$, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
cbr   r16,$F0      ; Clear upper nibble of r16
cbr   r18,1        ; Clear bit 0 in r18
```

Words: 1 (2 bytes)
Cycles: 1

CLC - Clear Carry Flag

Description:
Clears the Carry flag (C) in SREG (status register).

Operation:
(i) C ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLC	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1000	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	**0**

C: 0
Carry flag cleared

Example:

```
add r0,r0          ; Add r0 to itself
clc                ; Clear carry flag
```

Words: 1 (2 bytes)
Cycles: 1

CLI - Clear Global Interrupt Flag

Description:
Clears the Global Interrupt flag (I) in SREG (status register).

Operation:
(i) I ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLI	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1111	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
0	-	-	-	-	-	-	-

I: 0
Global Interrupt flag cleared

Example:

```
cli                 ; Disable interrupts
in    r11,$16       ; Read port B
sei                 ; Enable interrupts
```

Words: 1 (2 bytes)
Cycles: 1

CLN - Clear Negative Flag

Description:
Clears the Negative flag (N) in SREG (status register).

Operation:
(i) N ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLN	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1010	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	**0**	-	-

N: 0
Negative flag cleared

Example:

```
add   r2,r3       ; Add r3 to r2
cln               ; Clear negative flag
```

Words: 1 (2 bytes)
Cycles: 1

CLH - Clear Half Carry Flag

Description:
Clears the Half Carry flag (H) in SREG (status register).

Operation:
(i) H ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLH	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1101	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	**0**	-	-	-	-	-

H: 0
Half Carry flag cleared

Example:

```
clh          ; Clear the Half Carry flag
```

Words: 1 (2 bytes)
Cycles: 1

CLR - Clear Register

Description:
Clears a register. This instruction performs an Exclusive OR between a register and itself. This will clear all bits in the register.

Operation:
(i) Rd ← Rd ⊕ Rd

	Syntax:	Operands:	Program Counter:
(i)	CLR Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode: (see EOR Rd,Rd)

0010	01dd	dddd	dddd

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	0	0	0	1	-

S: 0
Cleared

V: 0
Cleared

N: 0
Cleared

Z: 1
Set

R (Result) equals Rd after the operation.

Example:

```
            clr   r18          ; clear r18
loop:       inc   r18          ; increase r18
            ...
            cpi   r18,$50      ; Compare r18 to $50
            brne  loop
```

Words: 1 (2 bytes)
Cycles: 1

CLS - Clear Signed Flag

Description:
Clears the Signed flag (S) in SREG (status register).

Operation:
(i) S ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLS	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1100	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	0	-	-	-	-

S: 0
Signed flag cleared

Example:

```
add   r2,r3        ; Add r3 to r2
cls                ; Clear signed flag
```

Words: 1 (2 bytes)
Cycles: 1

CLT - Clear T Flag

Description:
Clears the T flag in SREG (status register).

Operation:
(i) T ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLT	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1110	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	0	-	-	-	-	-	-

T: 0
T flag cleared

Example:
```
clt          ; Clear T flag
```

Words: 1 (2 bytes)
Cycles: 1

CLV - Clear Overflow Flag

Description:
Clears the Overflow flag (V) in SREG (status register).

Operation:
(i) V ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLV	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1011	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	**0**	-	-	-

V: 0
Overflow flag cleared

Example:

```
add   r2,r3        ; Add r3 to r2
clv                ; Clear overflow flag
```

Words: 1 (2 bytes)
Cycles: 1

CLZ - Clear Zero Flag

Description:
Clears the Zero flag (Z) in SREG (status register).

Operation:
(i) Z ← 0

	Syntax:	Operands:	Program Counter:
(i)	CLZ	None	PC ← PC + 1

16 bit Opcode:

1001	0100	1001	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	**0**	-

Z: 0
Zero flag cleared

Example:

```
add   r2,r3       ; Add r3 to r2
clz               ; Clear zero
```

Words: 1 (2 bytes)
Cycles: 1

COM - One's Complement

Description:
This instruction performs a one's complement of register Rd

Operation:
(i) Rd ← $FF - Rd

	Syntax:	Operands:	Program Counter:
(i)	COM Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	1

S: $N \oplus V$
For signed tests.

V: 0
Cleared.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; Cleared otherwise.

C: 1
Set.

R (Result) equals Rd after the operation.

Example:

```
            com   r4          ; Take one's complement of r4
            breq  zero        ; Branch if zero
            ...
zero:       nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

CP - Compare

Description:
This instruction performs a compare between two registers Rd and Rr. None of the registers are changed. All conditional branches can be used after this instruction.

Operation:
(i) Rd - Rr

	Syntax:	Operands:	Program Counter:
(i)	CP Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0001	01rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot Rr3 + Rr3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{Rr7} \cdot \overline{R7} + \overline{Rd7} \cdot Rr7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $\overline{Rd7} \cdot Rr7 + Rr7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of the contents of Rr is larger than the absolute value of Rd; cleared otherwise.

R (Result) after the operation.

Example:

```
        cp    r4,r19      ; Compare r4 with r19
        brne  noteq       ; Branch if r4 <> r19
        ...
noteq:  nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

CPC - Compare with Carry

Description:
This instruction performs a compare between two registers Rd and Rr and also takes into account the previous carry. None of the registers are changed. All conditional branches can be used after this instruction.

	Operation:		
(i)	Rd - Rr - C		
	Syntax:	**Operands:**	**Program Counter:**
(i)	CPC Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0000	01rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot Rr3 + Rr3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{Rr7} \cdot \overline{R7} + \overline{Rd7} \cdot Rr7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0} \cdot Z$
Previous value remains unchanged when the result is zero; cleared otherwise.

C: $\overline{Rd7} \cdot Rr7 + Rr7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of the contents of Rr plus previous carry is larger than the absolute value of Rd; cleared otherwise.

R (Result) after the operation.

Example:

```
                        ; Compare r3:r2 with r1:r0
        cp    r2,r0     ; Compare low byte
        cpc   r3,r1     ; Compare high byte
        brne  noteq     ; Branch if not equal
        ...
noteq:  nop             ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

CPI - Compare with Immediate

Description:
This instruction performs a compare between register Rd and a constant. The register is not changed. All conditional branches can be used after this instruction.

Operation:
(i) Rd - K

	Syntax:	Operands:	Program Counter:
(i)	CPI Rd,K	$16 \le d \le 31, 0 \le K \le 255$	PC ← PC + 1

16 bit Opcode:

0011	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot K3 + K3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{K7} \cdot \overline{R7} + \overline{Rd7} \cdot K7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $\overline{Rd7} \cdot K7 + K7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of K is larger than the absolute value of Rd; cleared otherwise.

R (Result) after the operation.

Example:

```
            cpi   r19,3       ; Compare r19 with 3
            brne  error       ; Branch if r19<>3
            ...
error:      nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

CPSE - Compare Skip if Equal

Description:
This instruction performs a compare between two registers Rd and Rr, and skips the next instruction if Rd = Rr.

Operation:
(i) If Rd = Rr then PC ← PC + 2 (or 3) else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	CPSE Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1, Condition false - no skip PC ← PC + 2, Skip a one word instruction PC ← PC + 3, Skip a two word instruction

16 bit Opcode:

0001	00rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
inc   r4          ; Increase r4
cpse  r4,r0       ; Compare r4 to r0
neg   r4          ; Only executed if r4<>r0
nop               ; Continue (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

DEC - Decrement

Description:
Subtracts one -1- from the contents of register Rd and places the result in the destination register Rd.

The C flag in SREG is not affected by the operation, thus allowing the DEC instruction to be used on a loop counter in multiple-precision computations.

When operating on unsigned values, only BREQ and BRNE branches can be expected to perform consistently. When operating on two's complement values, all signed branches are available.

Operation:
(i) Rd ← Rd - 1

	Syntax:	Operands:	Program Counter:
(i)	DEC Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	1010

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	-

S: $N \oplus V$
For signed tests.

V: $\overline{R7} \cdot R6 \cdot R5 \cdot R4 \cdot R3 \cdot R2 \cdot R1 \cdot R0$
Set if two's complement overflow resulted from the operation; cleared otherwise. Two's complement overflow occurs if and only if Rd was $80 before the operation.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; Cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
        ldi   r17,$10      ; Load constant in r17
loop:   add   r1,r2        ; Add r2 to r1
        dec   r17          ; Decrement r17
        brne  loop         ; Branch if r17<>0
        nop                ; Continue (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

EOR - Exclusive OR

Description:
Performs the logical EOR between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd ⊕ Rr

	Syntax:	Operands:	Program Counter:
(i)	EOR Rd,Rr	0 ≤ d ≤ 31, 0 ≤ r ≤ 31	PC ← PC + 1

16 bit Opcode:

0010	01rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
eor   r4,r4       ; Clear r4
eor   r0,r22      ; Bitwise exclusive or between r0 and r22
```

Words: 1 (2 bytes)
Cycles: 1

ICALL - Indirect Call to Subroutine

Description:
Indirect call of a subroutine pointed to by the Z (16 bits) pointer register in the register file. The Z pointer register is 16 bits wide and allows call to a subroutine within the current 64K words (128K bytes) section in the program memory space.

Operation:

(i) PC(15-0) ← Z(15 - 0) Devices with 16 bits PC, 128K bytes program memory maximum.

(ii) PC(15-0) ← Z(15 - 0) Devices with 22 bits PC, 8M bytes program memory maximum.
PC(21-16) is unchanged

	Syntax:	Operands:	Program Counter:	Stack
(i)	ICALL	None	See Operation	STACK ← PC+1 SP ← SP-2 (2 bytes, 16 bits)
(ii)	ICALL	None	See Operation	STACK ← PC+1 SP ← SP-3 (3 bytes, 22 bits)

16 bit Opcode:

1001	0101	XXXX	1001

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
mov   r30,r0      ; Set offset to call table
icall             ; Call routine pointed to by r31:r30
```

Words: 1 (2 bytes)
Cycles: 3

IJMP - Indirect Jump

Description:
Indirect jump to the address pointed to by the Z (16 bits) pointer register in the register file. The Z pointer register is 16 bits wide and allows jump within the current 64K words (128K bytes) section of program memory.

Operation:

(i) PC ← Z(15 - 0) Devices with 16 bits PC, 128K bytes program memory maximum.

(ii) PC(15-0) ← Z(15-0) Devices with 22 bits PC, 8M bytes program memory maximum.
PC(21-16) is unchanged

	Syntax:	Operands:	Program Counter:	Stack
(ii)	IJMP	None	See Operation	Not Affected
(iii)	IJMP	None	See Operation	Not Affected

16 bit Opcode:

1001	0100	XXXX	1001

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
mov   r30,r0      ; Set offset to jump table
ijmp              ; Jump to routine pointed to by r31:r30
```

Words: 1 (2 bytes)
Cycles: 2

IN - Load an I/O Port to Register

Description:
Loads data from the I/O Space (Ports, Timers, Configuration registers etc.) into register Rd in the register file.

Operation:
(i) Rd ← P

	Syntax:	Operands:	Program Counter:
(i)	IN Rd,P	$0 \le d \le 31, 0 \le P \le 63$	PC ← PC + 1

16 bit Opcode:

1011	0PPd	dddd	PPPP

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            in    r25,$16     ; Read Port B
            cpi   r25,4       ; Compare read value to constant
            breq  exit        ; Branch if r25=4
            ...
exit:       nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

INC - Increment

Description:
Adds one -1- to the contents of register Rd and places the result in the destination register Rd.

The C flag in SREG is not affected by the operation, thus allowing the INC instruction to be used on a loop counter in multiple-precision computations.

When operating on unsigned numbers, only BREQ and BRNE branches can be expected to perform consistently. When operating on two's complement values, all signed branches are available.

Operation:
(i) Rd ← Rd + 1

	Syntax:	Operands:	Program Counter:
(i)	INC Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0011

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	-

S: $N \oplus V$
For signed tests.

V: $R7 \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if two's complement overflow resulted from the operation; cleared otherwise. Two's complement overflow occurs if and only if Rd was $7F before the operation.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; Cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
        clr   r22          ; clear r22
loop:   inc   r22          ; increment r22
        ...
        cpi   r22,$4F      ; Compare r22 to $4f
        brne  loop         ; Branch if not equal
        nop                ; Continue (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

JMP - Jump

Description:
Jump to an address within the entire 4M (words) program memory. See also RJMP.

	Operation:
(i)	PC ← k

	Syntax:	**Operands:**	**Program Counter:**	**Stack**
(i)	JMP k	0 ≤ k ≤ 4M	PC ← k	Unchanged

32 bit Opcode:

1001 kkkk	010k kkkk	kkkk kkkk	110k kkkk

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            mov   r1,r0       ; Copy r0 to r1
            jmp   farplc      ; Unconditional jump
            ...
farplc:     nop               ; Jump destination (do nothing)
```

Words: 2 (4 bytes)
Cycles : 3

LD - Load Indirect from SRAM to Register using Index X

Description:
Loads one byte indirect from SRAM to register. The SRAM location is pointed to by the X (16 bits) pointer register in the register file. Memory access is limited to the current SRAM page of 64K bytes. To access another SRAM page the RAMPX in register in the I/O area has to be changed.

The X pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are especially suited for accessing arrays, tables, and stack pointer usage of the X pointer register.

Using the X pointer:

	Operation:		**Comment:**
(i)	Rd ← (X)		X: Unchanged
(ii)	Rd ← (X)	X ← X + 1	X: Post incremented
(iii)	X ← X - 1	Rd ← (X)	X: Pre decremented

	Syntax:	**Operands:**	**Program Counter:**
(i)	LD Rd, X	$0 \le d \le 31$	PC ← PC + 1
(ii)	LD Rd, X+	$0 \le d \le 31$	PC ← PC + 1
(iii)	LD Rd,-X	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode :

(i)	1001	000d	dddd	1100
(ii)	1001	000d	dddd	1101
(iii)	1001	000d	dddd	1110

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r27          ; Clear X high byte
ldi   r26,$20      ; Set X low byte to $20
ld    r0,X+        ; Load r0 with SRAM loc. $20(X post inc)
ld    r1,X         ; Load r1 with SRAM loc. $21
ldi   r26,$23      ; Set X low byte to $23
ld    r2,X         ; Load r2 with SRAM loc. $23
ld    r3,-X        ; Load r3 with SRAM loc. $22(X pre dec)
```

Words: 1 (2 bytes)
Cycles: 2

LD (LDD) - Load Indirect from SRAM to Register using Index Y

Description:
Loads one byte indirect with or without displacement from SRAM to register. The SRAM location is pointed to by the Y (16 bits) pointer register in the register file. Memory access is limited to the current SRAM page of 64K bytes. To access another SRAM page the RAMPY register in the I/O area has to be changed.

The Y pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are especially suited for accessing arrays, tables, and stack pointer usage of the Y pointer register.

Using the Y pointer:

	Operation:		**Comment:**
(i)	Rd ← (Y)		Y: Unchanged
(ii)	Rd ← (Y)	Y ← Y + 1	Y: Post incremented
(iii)	Y ← Y - 1	Rd ← (Y)	Y: Pre decremented
(iiii)	Rd ← (Y+q)		Y: Unchanged, q: Displacement

	Syntax:	**Operands:**	**Program Counter:**
(i)	LD Rd, Y	$0 \le d \le 31$	PC ← PC + 1
(ii)	LD Rd, Y+	$0 \le d \le 31$	PC ← PC + 1
(iii)	LD Rd,-Y	$0 \le d \le 31$	PC ← PC + 1
(iiii)	LDD Rd, Y+q	$0 \le d \le 31, 0 \le q \le 63$	PC ← PC + 1

16 bit Opcode :

(i)	1000	000d	dddd	1000
(ii)	1001	000d	dddd	1001
(iii)	1001	000d	dddd	1010
(iiii)	10q0	qq0d	dddd	1qqq

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r29          ; Clear Y high byte
ldi   r28,$20      ; Set Y low byte to $20
ld    r0,Y+        ; Load r0 with SRAM loc. $20(Y post inc)
ld    r1,Y         ; Load r1 with SRAM loc. $21
ldi   r28,$23      ; Set Y low byte to $23
ld    r2,Y         ; Load r2 with SRAM loc. $23
ld    r3,-Y        ; Load r3 with SRAM loc. $22(Y pre dec)
ldd   r4,Y+2       ; Load r4 with SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

LD (LDD) - Load Indirect From SRAM to Register using Index Z

Description:
Loads one byte indirectly with or without displacement from SRAM to register. The SRAM location is pointed to by the Z (16 bits) pointer register in the register file. Memory access is limited to the current SRAM page of 64K bytes. To access another SRAM page the RAMPZ register in the I/O area has to be changed.

The Z pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are especially suited for stack pointer usage of the Z pointer register, however because the Z pointer register can be used for indirect subroutine calls, indirect jumps and table lookup, it is often more convenient to use the X or Y pointer as a dedicated stack pointer.

For using the Z pointer for table lookup in program memory see the LPM instruction.

Using the Z pointer:

	Operation:		Comment:
(i)	Rd ← (Z)		Z: Unchanged
(ii)	Rd ← (Z)	Z ← Z + 1	Z: Post increment
(iii)	Z ← Z -1	Rd ← (Z)	Z: Pre decrement
(iiii)	Rd ← (Z+q)		Z: Unchanged, q: Displacement

	Syntax:	Operands:	Program Counter:
(i)	LD Rd, Z	0 ≤ d ≤ 31	PC ← PC + 1
(ii)	LD Rd, Z+	0 ≤ d ≤ 31	PC ← PC + 1
(iii)	LD Rd,-Z	0 ≤ d ≤ 31	PC ← PC + 1
(iiii)	LDD Rd, Z+q	0 ≤ d ≤ 31, 0 ≤ q ≤ 63	PC ← PC + 1

16 bit Opcode :

(i)	1000	000d	dddd	0000
(ii)	1001	000d	dddd	0001
(iii)	1001	000d	dddd	0010
(iiii)	10q0	qq0d	dddd	0qqq

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r31          ; Clear Z high byte
ldi   r30,$20      ; Set Z low byte to $20
ld    r0,Z+        ; Load r0 with SRAM loc. $20(Z post inc)
ld    r1,Z         ; Load r1 with SRAM loc. $21
ldi   r30,$23      ; Set Z low byte to $23
ld    r2,Z         ; Load r2 with SRAM loc. $23
ld    r3,-Z        ; Load r3 with SRAM loc. $22(Z pre dec)
ldd   r4,Z+2       ; Load r4 with SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

LDI - Load Immediate

Description:
Loads an 8 bit constant directly to register 16 to 31.

Operation:
(i) Rd ← K

	Syntax:	Operands:	Program Counter:
(i)	LDI Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC ← PC + 1

16 bit Opcode:

1110	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r31          ; Clear Z high byte
ldi   r30,$F0      ; Set Z low byte to $F0
lpm                ; Load constant from program
                   ; memory pointed to by Z
```

Words: 1 (2 bytes)
Cycles: 1

LPM - Load Program Memory

Description:
Loads one byte pointed to by the Z register into register 0 (R0). This instruction features a 100% space effective constant initialization or constant data fetch. The program memory is organized in 16 bits words and the LSB of the Z (16 bits) pointer selects either low byte (0) or high byte (1). This instruction can address the first 64K bytes (32K words) of program memory.

	Operation:	Comment:
(i)	R0 ← (Z)	Z points to program memory

	Syntax:	Operands:	Program Counter:
(i)	LPM	None	PC ← PC + 1

16 bit Opcode:

1001	0101	110X	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r31          ; Clear Z high byte
ldi   r30,$F0      ; Set Z low byte
lpm                ; Load constant from program
                   ; memory pointed to by Z (r31:r30)
```

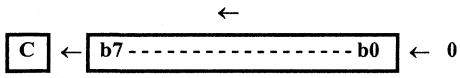
Words: 1 (2 bytes)
Cycles: 3

LSL - Logical Shift Left

Description:
Shifts all bits in Rd one place to the left. Bit 0 is cleared. Bit 7 is loaded into the C flag of the SREG. This operation effectively multiplies an unsigned value by two.

Operation:

(i)



	Syntax:	Operands:	Program Counter:
(i)	LSL Rd	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode: (see ADD Rd,Rd)

0000	11dd	dddd	dddd

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S: $N \oplus V$, For signed tests.

V: $N \oplus C$ (For N and C after the shift)
Set if (N is set and C is clear) or (N is clear and C is set); Cleared otherwise (for values of N and C after the shift).

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: Rd7
Set if, before the shift, the MSB of Rd was set; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

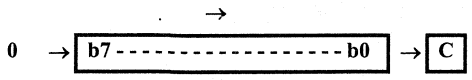
```
add   r0,r4        ; Add r4 to r0
lsl   r0           ; Multiply r0 by 2
```

Words: 1 (2 bytes)
Cycles: 1

LSR - Logical Shift Right

Description:
Shifts all bits in Rd one place to the right. Bit 7 is cleared. Bit 0 is loaded into the C flag of the SREG. This operation effectively divides an unsigned value by two. The C flag can be used to round the result.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	LSR Rd	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0110

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	0	⇔	⇔

S: $N \oplus V$, For signed tests.

V: $N \oplus C$ (For N and C after the shift)
Set if (N is set and C is clear) or (N is clear and C is set); Cleared otherwise (for values of N and C after the shift).

N: 0

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: Rd0
Set if, before the shift, the LSB of Rd was set; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
add   r0,r4        ; Add r4 to r0
lsr   r0           ; Divide r0 by 2
```

Words: 1 (2 bytes)
Cycles: 1

MOV - Copy Register

Description:
This instruction makes a copy of one register into another. The source register Rr is left unchanged, while the destination register Rd is loaded with a copy of Rr.

Operation:
(i) Rd ← Rr

	Syntax:	Operands:	Program Counter:
(i)	MOV Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0010	11rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

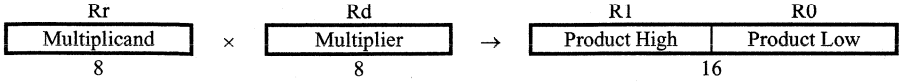
Example:

```
            mov   r16,r0      ; Copy r0 to r16
            call  check       ; Call subroutine
            ...
check:      cpi   r16,$11     ; Compare r16 to $11
            ...
            ret               ; Return from subroutine
```

Words: 1 (2 bytes)
Cycles: 1

MUL - Multiply

Description:
This instruction performs 8-bit × 8-bit → 16-bit unsigned multiplication.



The multiplicand Rr and the multiplier Rd are two registers. The 16-bit product is placed in R1 (high byte) and R0 (low byte). Note that if the multiplicand and the multiplier is selected from R0 or R1 the result will overwrite those after multiplication.

Operation:
(i) R1,R0 ← Rr × Rd

	Syntax:	Operands:	Program Counter:
(i)	MUL Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

1001	11rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	⇔

C: R15
Set if bit 15 of the result is set; cleared otherwise.

R (Result) equals R1,R0 after the operation.

Example:

```
mul   r6,r5 ; Multiply r6 and r5
mov   r6,r1 ; Copy result back in r6:r5
mov   r5,r0 ; Copy result back in r6:r5
```

Words: 1 (2 bytes)
Cycles: 2

Not available in base-line microcontrollers.

NEG - Two's Complement

Description:
Replaces the contents of register Rd with its two's complement; the value $80 is left unchanged.

Operation:
(i) Rd ← $00 - Rd

	Syntax:	Operands:	Program Counter:
(i)	NEG Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0001

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$
For signed tests.

V: $R7 \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if there is a two's complement overflow from the implied subtraction from zero; cleared otherwise. A two's complement overflow will occur if and only if the contents of the Register after operation (Result) is $80.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; Cleared otherwise.

C: R7 + R6 + R5 + R4 + R3 + R2 + R1 + R0
Set if there is a borrow in the implied subtraction from zero; cleared otherwise. The C flag will be set in all cases except when the contents of Register after operation is $00.

R (Result) equals Rd after the operation.

Example:

```
            sub   r11,r0      ; Subtract r0 from r11
            brpl  positive    ; Branch if result positive
            neg   r11         ; Take two's complement of r11
positive:   nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

NOP - No Operation

Description:
This instruction performs a single cycle No Operation.

	Operation:
(i)	No

	Syntax:	Operands:	Program Counter:
(i)	NOP	None	PC ← PC + 1

16 bit Opcode:

0000	0000	0000	0000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r16          ; Clear r16
ser   r17          ; Set r17
out   $18,r16      ; Write zeros to Port B
nop                ; Wait (do nothing)
out   $18,r17      ; Write ones to Port B
```

Words: 1 (2 bytes)
Cycles: 1

OR - Logical OR

Description:
Performs the logical OR between the contents of register Rd and register Rr and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd v Rr

	Syntax:	Operands:	Program Counter:
(i)	OR Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0010	10rd	dddd	rrrr

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
        or    r15,r16      ; Do bitwise or between registers
        bst   r15,6        ; Store bit 6 of r15 in T flag
        brts  ok           ; Branch if T flag set
        ...
ok:     nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

ORI - Logical OR with Immediate

Description:
Performs the logical OR between the contents of register Rd and a constant and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd v K

	Syntax:	Operands:	Program Counter:
(i)	ORI Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC ← PC + 1

16 bit Opcode:

0110	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
ori   r16,$F0      ; Set high nibble of r16
ori   r17,1        ; Set bit 0 of r17
```

Words: 1 (2 bytes)
Cycles: 1

OUT - Store Register to I/O port

Description:
Stores data from register Rr in the register file to I/O space (Ports, Timers, Configuration registers etc.).

	Operation:
(i)	P ← Rr

	Syntax:	Operands:	Program Counter:
(i)	OUT P,Rr	$0 \le r \le 31, 0 \le P \le 63$	PC ← PC + 1

16 bit Opcode:

1011	1PPr	rrrr	PPPP

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r16          ; Clear r16
ser   r17          ; Set r17
out   $18,r16      ; Write zeros to Port B
nop                ; Wait (do nothing)
out   $18,r17      ; Write ones to Port B
```

Words: 1 (2 bytes)
Cycles: 1

POP - Pop Register from Stack

Description:
This instruction loads register Rd with a byte from the STACK.

	Operation:
(i)	Rd ← STACK

	Syntax:	Operands:	Program Counter:	Stack
(i)	POP Rd	0 ≤ d ≤ 31	PC ← PC + 1	SP ← SP + 1

16 bit Opcode:

1001	000d	dddd	1111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            call  routine     ; Call subroutine
            ...
routine:    push  r14         ; Save r14 on the stack
            push  r13         ; Save r13 on the stack
            ...
            pop   r13         ; Restore r13
            pop   r14         ; Restore r14
            ret               ; Return from subroutine
```

Words: 1 (2 bytes)
Cycles: 2

PUSH - Push Register on Stack

Description:
This instruction stores the contents of register Rr on the STACK.

Operation:
(i) STACK ← Rr

	Syntax:	Operands:	Program Counter:	Stack:
(i)	PUSH Rr	0 ≤ r ≤ 31	PC ← PC + 1	SP ← SP - 1

16 bit Opcode:

1001	001d	dddd	1111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            call   routine      ; Call subroutine
            ...
routine:    push   r14          ; Save r14 on the stack
            push   r13          ; Save r13 on the stack
            ...
            pop    r13          ; Restore r13
            pop    r14          ; Restore r14
            ret                 ; Return from subroutine
```

Words: 1 (2 bytes)
Cycles: 2

RCALL - Relative Call to Subroutine

Description:
Calls a subroutine within ± 2K words (4K bytes). The return address (the instruction after the RCALL) is stored onto the stack. (See also CALL).

Operation:

(i) PC ← PC + k + 1 Devices with 16 bits PC, 128K bytes program memory maximum.
(ii) PC ← PC + k + 1 Devices with 22 bits PC, 8M bytes program memory maximum.

	Syntax:	Operands:	Program Counter:	Stack
(i)	RCALL k	-2K ≤ k ≤ 2K	PC ← PC + k + 1	STACK ← PC+1 SP ← SP-2 (2 bytes, 16 bits)
(ii)	RCALL k	-2K ≤ k ≤ 2K	PC ← PC + k + 1	STACK ← PC+1 SP ← SP-3 (3 bytes, 22 bits)

16 bit Opcode:

1101	kkkk	kkkk	kkkk

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            rcall  routine      ; Call subroutine
            ...
routine:    push   r14          ; Save r14 on the stack
            ...
            pop    r14          ; Restore r14
            ret                 ; Return from subroutine
```

Words: 1 (2 bytes)
Cycles: 3

RET - Return from Subroutine

Description:
Returns from subroutine. The return address is loaded from the STACK.

Operation:

(i) PC(15-0) ← STACK Devices with 16 bits PC, 128K bytes program memory maximum.
(ii) PC(21-0) ← STACK Devices with 22 bits PC, 8M bytes program memory maximum.

	Syntax:	Operands:	Program Counter:	Stack
(i)	RET	None	See Operation	SP ← SP +2, (2 bytes,16 bits pulled)
(ii)	RET	None	See Operation	SP ← SP +3, (3 bytes, 22 bits pulled)

16 bit Opcode:

1001	0101	0XX0	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
            call  routine     ; Call subroutine
            ...
routine:    push  r14         ; Save r14 on the stack
            ...
            pop   r14         ; Restore r14
            ret               ; Return from subroutine
```

Words: 1 (2 bytes)
Cycles: 4

RETI - Return from Interrupt

Description:
Returns from interrupt. The return address is loaded from the STACK and the global interrupt flag is set.

Operation:

(i) PC(15-0) ← STACK Devices with 16 bits PC, 128K bytes program memory maximum.
(ii) PC(21-0) ← STACK Devices with 22 bits PC, 8M bytes program memory maximum.

	Syntax:	Operands:	Program Counter:	Stack
(i)	RETI	None	See Operation	SP ← SP +2 (2 bytes, 16 bits)
(ii)	RETI	None	See Operation	SP ← SP +3 (3 bytes, 22 bits)

16 bit Opcode:

1001	0101	0XX1	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
1	-	-	-	-	-	-	-

I: 1
The I flag is set.

Example:

```
            ...
extint:     push  r0            ; Save r0 on the stack
            ...
            pop r0              ; Restore r0
            reti                ; Return and enable interrupts
```

Words: 1 (2 bytes)
Cycles: 4

RJMP - Relative Jump

Description:
Relative jump to an address within PC-2K and PC + 2K (words). In the assembler labels, are used instead of relative operands. For AVR microcontrollers with smaller memory than 4K words (8K bytes) this instruction can address the entire memory from every address location.

Operation:
(i) PC ← PC + k + 1

	Syntax:	Operands:	Program Counter:	Stack
(i)	RJMP k	-2K ≤ k ≤ 2K	PC ← PC + k + 1	Unchanged

16 bit Opcode:

1100	kkkk	kkkk	kkkk

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

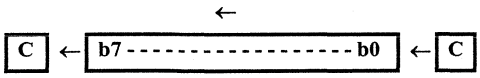
```
            cpi   r16,$42     ; Compare r16 to $42
            brne  error       ; Branch if r16 <> $42
            rjmp  ok          ; Unconditional branch
error:      add   r16,r17     ; Add r17 to r16
            inc   r16         ; Increment r16
ok:         nop               ; Destination for rjmp (do nothing)
```

Words: 1 (2 bytes)
Cycles: 2

ROL - Rotate Left trough Carry

Description:
Shifts all bits in Rd one place to the left. The C flag is shifted into bit 0 of Rd. Bit 7 is shifted into the C flag.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	ROL Rd	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode: (see ADC Rd,Rd)

0001	11dd	dddd	dddd

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S: $N \oplus V$, For signed tests.

V: $N \oplus C$ (For N and C after the shift)
Set if (N is set and C is clear) or (N is clear and C is set); Cleared otherwise (for values of N and C after the shift).

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: Rd7
Set if, before the shift, the MSB of Rd was set; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

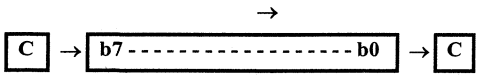
```
            rol   r15          ; Rotate left
            brcs  oneenc       ; Branch if carry set
            ...
oneenc:     nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

ROR - Rotate Right trough Carry

Description:
Shifts all bits in Rd one place to the right. The C flag is shifted into bit 7 of Rd. Bit 0 is shifted into the C flag.

Operation:



	Syntax:	Operands:	Program Counter:
(i)	ROR Rd	$0 \le d \le 31$	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	⇔	⇔	⇔	⇔

S: $N \oplus V$, For signed tests.

V: $N \oplus C$ (For N and C after the shift)
Set if (N is set and C is clear) or (N is clear and C is set); Cleared otherwise (for values of N and C after the shift).

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: Rd0
Set if, before the shift, the LSB of Rd was set; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
            ror   r15          ; Rotate right
            brcc  zeroenc      ; Branch if carry cleared
            ...
zeroenc:    nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

SBC - Subtract with Carry

Description:
Subtracts two registers and subtracts with the C flag and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd - Rr - C

	Syntax:	Operands:	Program Counter:
(i)	SBC Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0000	10rd	dddd	rrrr

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot Rr3 + Rr3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{Rr7} \cdot \overline{R7} + \overline{Rd7} \cdot Rr7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0} \cdot Z$
Previous value remains unchanged when the result is zero; cleared otherwise.

C: $\overline{Rd7} \cdot Rr7 + Rr7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of the contents of Rr plus previous carry is larger than the absolute value of the Rd; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
                        ; Subtract r1:r0 from r3:r2
sub     r2,r0           ; Subtract low byte
sbc     r3,r1           ; Subtract with carry high byte
```

Words: 1 (2 bytes)
Cycles: 1

SBCI - Subtract Immediate with Carry

Description:
Subtracts a constant from a register and subtracts with the C flag and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd - K - C

	Syntax:	Operands:	Program Counter:
(i)	SBCI Rd,K	$16 \le d \le 31, 0 \le K \le 255$	PC ← PC + 1

16 bit Opcode:

0100	KKKK	dddd	KKKK

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot K3 + K3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{K7} \cdot \overline{R7} + \overline{Rd7} \cdot K7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0} \cdot Z$
Previous value remains unchanged when the result is zero; cleared otherwise.

C: $\overline{Rd7} \cdot K7 + K7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of the constant plus previous carry is larger than the absolute value of Rd; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
                        ; Subtract $4F23 from r17:r16
subi   r16,$23          ; Subtract low byte
sbci   r17,$4F          ; Subtract with carry high byte
```

Words: 1 (2 bytes)
Cycles: 1

SBR - Set Bits in Register

Description:
Sets specified bits in register Rd. Performs the logical ORI between the contents of register Rd and a constant mask K and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd v K

	Syntax:	Operands:	Program Counter:
(i)	SBR Rd,K	16 ≤ d ≤ 31, 0 ≤ K ≤ 255	PC ← PC + 1

16 bit Opcode:

0110	KKKK	dddd	KKKK

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
sbr   r16,3       ; Set bits 0 and 1 in r16
sbr   r17,$F0     ; Set 4 MSB in r17
```

Words: 1 (2 bytes)
Cycles: 1

SBRC - Skip if Bit in Register is Cleared

Description:
This instruction tests a single bit in a register and skips the next instruction if the bit is cleared.

Operation:
(i) If Rr(b) = 0 then PC ← PC + 2 (or 3) else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	SBRC Rr,b	$0 \le r \le 31, 0 \le b \le 7$	PC ← PC + 1, If condition is false, no skip. PC ← PC + 2, If next instruction is one word. PC ← PC + 3, If next instruction is JMP or CALL

16 bit Opcode:

1111	110r	rrrr	Xbbb

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
sub   r0,r1       ; Subtract r1 from r0
sbrc  r0,7        ; Skip if bit 7 in r0 cleared
sub   r0,r1       ; Only executed if bit 7 in r0 not cleared
nop               ; Continue (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false (no skip)
2 if condition is true (skip is executed)

SBRS - Skip if Bit in Register is Set

Description:
This instruction tests a single bit in a register and skips the next instruction if the bit is set.

Operation:
(i) If Rr(b) = 1 then PC ← PC + 2 (or 3) else PC ← PC + 1

	Syntax:	Operands:	Program Counter:
(i)	SBRS Rr,b	$0 \le r \le 31$, $0 \le b \le 7$	PC ← PC + 1, Condition false - no skip PC ← PC + 2, Skip a one word instruction PC ← PC + 3, Skip a JMP or a CALL

16 bit Opcode:

1111	111r	rrrr	Xbbb

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
sub   r0,r1        ; Subtract r1 from r0
sbrs  r0,7         ; Skip if bit 7 in r0 set
neg   r0           ; Only executed if bit 7 in r0 not set
nop                ; Continue (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1 if condition is false (no skip)
2 if condition is true (skip is executed)

SEC - Set Carry Flag

Description:
Sets the Carry flag (C) in SREG (status register).

Operation:
(i) C ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEC	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0000	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	1

C: 1
Carry flag set

Example:

```
sec                 ; Set carry flag
adc   r0,r1         ; r0=r0+r1+1
```

Words: 1 (2 bytes)
Cycles: 1

SEI - Set Global Interrupt Flag

Description:
Sets the Global Interrupt flag (I) in SREG (status register).

Operation:
(i) I ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEI	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0111	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
1	-	-	-	-	-	-	-

I: 1
Global Interrupt flag set

Example:

```
cli                 ; Disable interrupts
in    r13,$16       ; Read Port B
sei                 ; Enable interrupts
```

Words: 1 (2 bytes)
Cycles: 1

SEN - Set Negative Flag

Description:
Sets the Negative flag (N) in SREG (status register).

Operation:
(i) N ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEN	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0010	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	1	-	-

N: 1
Negative flag set

Example:

```
add r2,r19        ; Add r19 to r2
sen               ; Set negative flag
```

Words: 1 (2 bytes)
Cycles: 1

SEH - Set Half Carry Flag

Description:
Sets the Half Carry (H) in SREG (status register).

Operation:
(i) H ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEH	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0101	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	1	-	-	-	-	-

H: 1
Half Carry flag set

Example:

```
seh                ; Set Half Carry flag
```

Words: 1 (2 bytes)
Cycles: 1

SER - Set all bits in Register

Description:
Loads $FF directly to register Rd.

Operation:
(i) Rd ← $FF

	Syntax:	Operands:	Program Counter:
(i)	SER Rd	16 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1110	1111	dddd	1111

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r16          ; Clear r16
ser   r17          ; Set r17
out   $18,r16      ; Write zeros to Port B
nop                ; Delay (do nothing)
out   $18,r17      ; Write ones to Port B
```

Words: 1 (2 bytes)
Cycles: 1

SES - Set Signed Flag

Description:
Sets the Signed flag (S) in SREG (status register).

Operation:
(i) S ← 1

	Syntax:	Operands:	Program Counter:
(i)	SES	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0100	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	1	-	-	-	-

S: 1
Signed flag set

Example:

```
add r2,r19          ; Add r19 to r2
ses                 ; Set negative flag
```

Words: 1 (2 bytes)
Cycles: 1

SET - Set T Flag

Description:
Sets the T flag in SREG (status register).

Operation:
(i) T ← 1

	Syntax:	Operands:	Program Counter:
(i)	SET	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0110	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	1	-	-	-	-	-	-

T: 1
T flag set

Example:

```
set                     ; Set T flag
```

Words: 1 (2 bytes)
Cycles: 1

SEV - Set Overflow Flag

Description:
Sets the Overflow flag (V) in SREG (status register).

Operation:
(i) V ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEV	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0011	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	1	-	-	-

V: 1
Overflow flag set

Example:

```
add r2,r19          ; Add r19 to r2
sev                 ; Set overflow flag
```

Words: 1 (2 bytes)
Cycles: 1

SEZ - Set Zero Flag

Description:
Sets the Zero flag (Z) in SREG (status register).

Operation:
(i) Z ← 1

	Syntax:	Operands:	Program Counter:
(i)	SEZ	None	PC ← PC + 1

16 bit Opcode:

1001	0100	0001	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	1	-

Z: 1
Zero flag set

Example:

```
add r2,r19          ; Add r19 to r2
sez                 ; Set zero flag
```

Words: 1 (2 bytes)
Cycles: 1

SLEEP

Description:
This instruction sets the circuit in sleep mode defined by the MCU control register. When an interrupt wakes up the MCU from a sleep state, the instruction following the SLEEP instruction will be executed before the interrupt handler is executed.

Operation:

Syntax:	Operands:	Program Counter:
SLEEP	None	PC ← PC + 1

16 bit Opcode:

1001	0101	100X	1000

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
mov   r0,r11      ; Copy r11 to r0
sleep             ; Put MCU in sleep mode
```

Words: 1 (2 bytes)
Cycles: 1

ST - Store Indirect From Register to SRAM using Index X

Description:
Stores one byte indirect from SRAM to register. The SRAM location is pointed to by the X (16 bits) pointer register in the register file. Memory access is limited to the current SRAM Page of 64K bytes. To access another SRAM page the RAMPX register in the I/O area has to be changed.

The X pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are especially suited for stack pointer usage of the X pointer register.

Using the X pointer:

	Operation:		**Comment:**
(i)	(X) ← Rr		X: Unchanged
(ii)	(X) ← Rr	X ← X+1	X: Post incremented
(iii)	X ← X - 1	(X) ← Rr	X: Pre decremented

	Syntax:	**Operands:**	**Program Counter:**
(i)	ST X, Rr	$0 \le r \le 31$	PC ← PC + 1
(ii)	ST X+, Rr	$0 \le r \le 31$	PC ← PC + 1
(iii)	ST -X, Rr	$0 \le r \le 31$	PC ← PC + 1

16 bit Opcode :

(i)	1001	001r	rrrr	1100
(ii)	1001	001r	rrrr	1101
(iii)	1001	001r	rrrr	1110

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r27          ; Clear X high byte
ldi   r26,$20      ; Set X low byte to $20
st    X+,r0        ; Store r0 in SRAM loc. $20(X post inc)
st    X,r1         ; Store r1 in SRAM loc. $21
ldi   r26,$23      ; Set X low byte to $23
st    r2,X         ; Store r2 in SRAM loc. $23
st    r3,-X        ; Store r3 in SRAM loc. $22(X pre dec)
```

Words: 1 (2 bytes)
Cycles: 2

ST (STD) - Store Indirect From Register to SRAM using Index Y

Description:
Stores one byte indirect with or without displacement from SRAM to register. The SRAM location is pointed to by the Y (16 bits) pointer register in the register file. Memory access is limited to the current SRAM Page of 64K bytes. To access another SRAM page the RAMPY register in the I/O area has to be changed.

The Y pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are especially suited for stack pointer usage of the Y pointer register.

Using the Y pointer:

	Operation:		Comment:
(i)	(Y) ← Rr		Y: Unchanged
(ii)	(Y) ← Rr	Y ← Y+1	Y: Post incremented
(iii)	Y ← Y - 1	(Y) ← Rr	Y: Pre decremented
(iiii)	(Y+q) ← Rr		Y: Unchanged, q: Displacement

	Syntax:	Operands:	Program Counter:
(i)	ST Y, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(ii)	ST Y+, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(iii)	ST -Y, Rr	0 ≤ r ≤ 31	PC ← PC + 1
(iiii)	STD Y+q, Rr	0 ≤ r ≤ 31, 0 ≤ q ≤ 63	PC ← PC + 1

16 bit Opcode :

(i)	1000	001r	rrrr	1000
(ii)	1001	001r	rrrr	1001
(iii)	1001	001r	rrrr	1010
(iiii)	10q0	qq1r	rrrr	1qqq

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r29          ; Clear Y high byte
ldi   r28,$20      ; Set Y low byte to $20
st    Y+,r0        ; Store r0 in SRAM loc. $20(Y post inc)
st    Y,r1         ; Store r1 in SRAM loc. $21
ldi   r28,$23      ; Set Y low byte to $23
st    Y,r2         ; Store r2 in SRAM loc. $23
st    -Y,r3        ; Store r3 in SRAM loc. $22(Y pre dec)
std   Y+2,r4       ; Store r4 in SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

ST (STD) - Store Indirect From Register to SRAM using Index Z

Description:
Stores one byte indirect with or without displacement from Register to SRAM. The SRAM location is pointed to by the Z (16 bits) pointer register in the register file. Memory access are limited to the current SRAM Page of 64K bytes. To access another SRAM page the RAMPZ register in the I/O area has to be changed.

The Z pointer register can either be left unchanged after the operation, or it can be incremented or decremented. These features are very suited for stack pointer usage of the Z pointer register, but because the Z pointer register can be used for indirect subroutine calls, indirect jumps and table lookup it is often more convenient to use the X or Y pointer as a dedicated stack pointer.

Using the Z pointer:

	Operation:		**Comment:**
(i)	(Z) ← Rr		Z: Unchanged
(ii)	(Z) ← Rr	Z ← Z+1	Z: Post incremented
(iii)	Z ← Z - 1	(Z) ← Rr	Z: Pre decremented
(iiii)	(Z+q) ← Rr		Z: Unchanged, q: Displacement

	Syntax:	**Operands:**	**Program Counter:**
(i)	ST Z, Rr	$0 \le r \le 31$	PC ← PC + 1
(ii)	ST Z+, Rr	$0 \le r \le 31$	PC ← PC + 1
(iii)	ST -Z, Rr	$0 \le r \le 31$	PC ← PC + 1
(iiii)	STD Z+q, Rr	$0 \le r \le 31, 0 \le q \le 63$	PC ← PC + 1

16 bit Opcode :

(i)	1000	001r	rrrr	0000
(ii)	1001	001r	rrrr	0001
(iii)	1001	001r	rrrr	0010
(iiii)	10q0	qq1r	rrrr	0qqq

Status Register (SREG) and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
clr   r31          ; Clear Z high byte
ldi   r30,$20      ; Set Z low byte to $20
st    Z+,r0        ; Store r0 in SRAM loc. $20(Z post inc)
st    Z,r1         ; Store r1 in SRAM loc. $21
ldi   r30,$23      ; Set Z low byte to $23
st    Z,r2         ; Store r2 in SRAM loc. $23
st    -Z,r3        ; Store r3 in SRAM loc. $22(Z pre dec)
std   Z+2,r4       ; Store r4 in SRAM loc. $24
```

Words: 1 (2 bytes)
Cycles: 2

SUB - Subtract without Carry

Description:
Subtracts two registers and places the result in the destination register Rd.

Operation:
(i) Rd ← Rd - Rr

	Syntax:	Operands:	Program Counter:
(i)	SUB Rd,Rr	$0 \le d \le 31, 0 \le r \le 31$	PC ← PC + 1

16 bit Opcode:

0001	10rd	dddd	rrrr

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot Rr3 + Rr3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{Rr7} \cdot \overline{R7} + \overline{Rd7} \cdot Rr7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $\overline{Rd7} \cdot Rr7 + Rr7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of the contents of Rr is larger than the absolute value of Rd; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
            sub   r13,r12      ; Subtract r12 from r13
            brne  noteq        ; Branch if r12<>r13
            ...
noteq:      nop                ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

SUBI - Subtract Immediate

Description:
Subtracts a register and a constant and places the result in the destination register Rd. This instruction is working on Register R16 to R31 and is very well suited for operations on the X, Y and Z pointers.

Operation:
(i) Rd ← Rd - K

	Syntax:	Operands:	Program Counter:
(i)	SUBI Rd,K	$16 \le d \le 31, 0 \le K \le 255$	PC ← PC + 1

16 bit Opcode:

0101	KKKK	dddd	KKKK

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	⇔	⇔	⇔	⇔	⇔	⇔

H: $\overline{Rd3} \cdot K3 + K3 \cdot R3 + R3 \cdot \overline{Rd3}$
Set if there was a borrow from bit 3; cleared otherwise

S: $N \oplus V$, For signed tests.

V: $Rd7 \cdot \overline{K7} \cdot \overline{R7} + \overline{Rd7} \cdot K7 \cdot R7$
Set if two's complement overflow resulted from the operation; cleared otherwise.

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

C: $\overline{Rd7} \cdot K7 + K7 \cdot R7 + R7 \cdot \overline{Rd7}$
Set if the absolute value of K is larger than the absolute value of Rd; cleared otherwise.

R (Result) equals Rd after the operation.

Example:

```
        subi   r22,$11      ; Subtract $11 from r22
        brne   noteq        ; Branch if r22<>$11
        ...
noteq:  nop                 ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

SWAP - Swap Nibbles

Description:
Swaps high and low nibbles in a register.

Operation:
(i) R(7-4) ← Rd(3-0), R(3-0) ← Rd(7-4)

	Syntax:	Operands:	Program Counter:
(i)	SWAP Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

1001	010d	dddd	0010

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

R (Result) equals Rd after the operation.

Example:

```
inc   r1          ; Increment r1
swap  r1          ; Swap high and low nibble of r1
inc   r1          ; Increment high nibble of r1
swap  r1          ; Swap back
```

Words: 1 (2 bytes)
Cycles: 1

TST - Test for Zero or Minus

Description:
Tests if a register is zero or negative. Performs a logical AND between a register and itself. The register will remain unchanged.

Operation:
(i) Rd ← Rd • Rd

	Syntax:	Operands:	Program Counter:
(i)	TST Rd	0 ≤ d ≤ 31	PC ← PC + 1

16 bit Opcode:

0010	00dd	dddd	dddd

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	⇔	0	⇔	⇔	-

S: N ⊕ V, For signed tests.

V: 0
Cleared

N: R7
Set if MSB of the result is set; cleared otherwise.

Z: $\overline{R7} \cdot \overline{R6} \cdot \overline{R5} \cdot \overline{R4} \cdot \overline{R3} \cdot \overline{R2} \cdot \overline{R1} \cdot \overline{R0}$
Set if the result is $00; cleared otherwise.

R (Result) equals Rd.

Example:

```
        tst   r0          ; Test r0
        breq  zero        ; Branch if r0=0
        ...
zero:   nop               ; Branch destination (do nothing)
```

Words: 1 (2 bytes)
Cycles: 1

WDR - Watchdog Reset

Description:
This instruction resets the Watchdog Timer. This instruction must be executed within a limited time given by the WD prescaler. See the Watchdog Timer hardware specification.

Operation:
(i) WD timer restart.

	Syntax:	Operands:	Program Counter:
(i)	WDR	None	PC ← PC + 1

16 bit Opcode:

1001	0101	101X	1000

Status Register and Boolean Formulae:

I	T	H	S	V	N	Z	C
-	-	-	-	-	-	-	-

Example:

```
wdr                ; Reset watchdog timer
```

Words: 1 (2 bytes)
Cycles: 1

5

AVR Development Tools

This section describes some of the development tools that are available for the 8-bit *AVR* family.

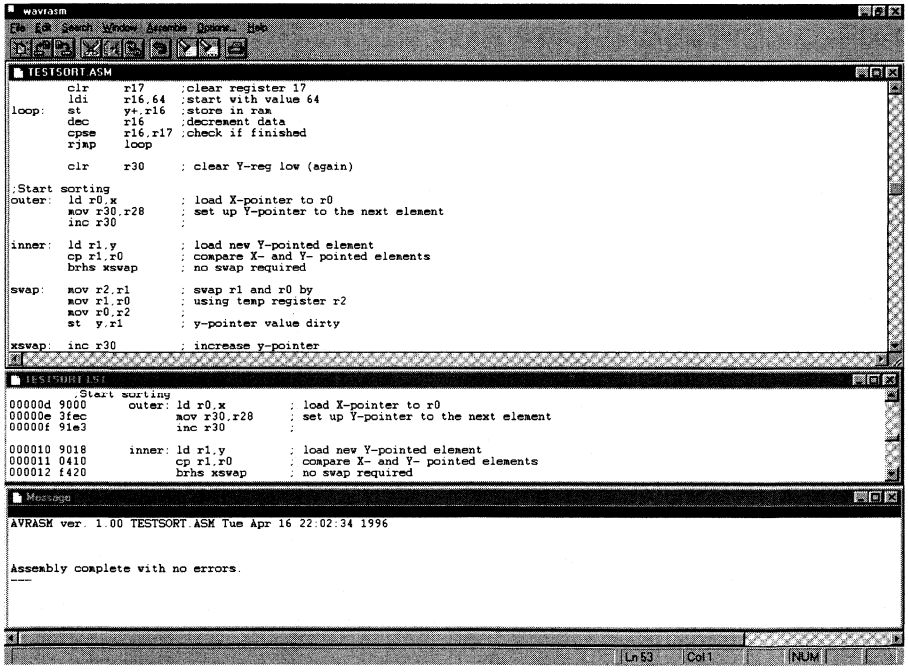
- **Atmel AVR Assembler**
- **Atmel AVR Simulator**
- **IAR ANSI C-Compiler, Assembler, Linker, Librarian & Debugger**
- **Atmel In-Circuit Emulator (ICE)**
- **EQUINOX Micro-Pro AVR Device Programmer**

There are a lot of development tools under development, please contact Atmel for more details.

Atmel *AVR* Assembler

Features:

- Translates Assembler source programs into object code
- Extremely fast assembling
- Supports all the microcontrollers in the AT90S family
- Powerful Macro capabilities
- Supports all standard output formats
- Easy to use MS-Windows interface
- Editor included in MS-Windows version
- Jump to next/previous error
- Also available in MS-DOS command line version



Powerful Macro Capabilities

The Assembler contains powerful macro capabilities, enabling the user to build a virtual instruction set which is structures of ordinary *AVR* instructions. For example, this macro does a 16 bit subtraction:

```
;
; SUB16 macro definition
```

```
; The macro subtracts a 16 bit constant from a register
; pair. A call to the macro is done by
; SUB16 Regh,Regl,Const
;
.MACRO SUB16                    ; Macro name
      subi  $1,low($2)          ; subtract low byte
      sbci  $0,high($2)         ; subtract high byte
.ENDMACRO

;      ...

; Call the macro
      ldi   r16,low(0x3400)     ; set values in registers
      ldi   r17,low(0x3400)     ;
      SUB16 r17,r16,0x23a0      ; compute 0x3400-0x23a0
```

Assembly Directives

The assembler supports a number of directives making the application development easier. In addition to the directives for macro generation and control, the assembler contains directives for:

- Including files. Included files can be nested.
- Set program origin.
- Symbol usage. The user can define symbols and labels and refer to these throughout the assembly program.
- Constant data initialization. The user can do constant initialization. Constants will be placed in the Flash program memory.
- List file control.
- Support of expressions in a C-like syntax.

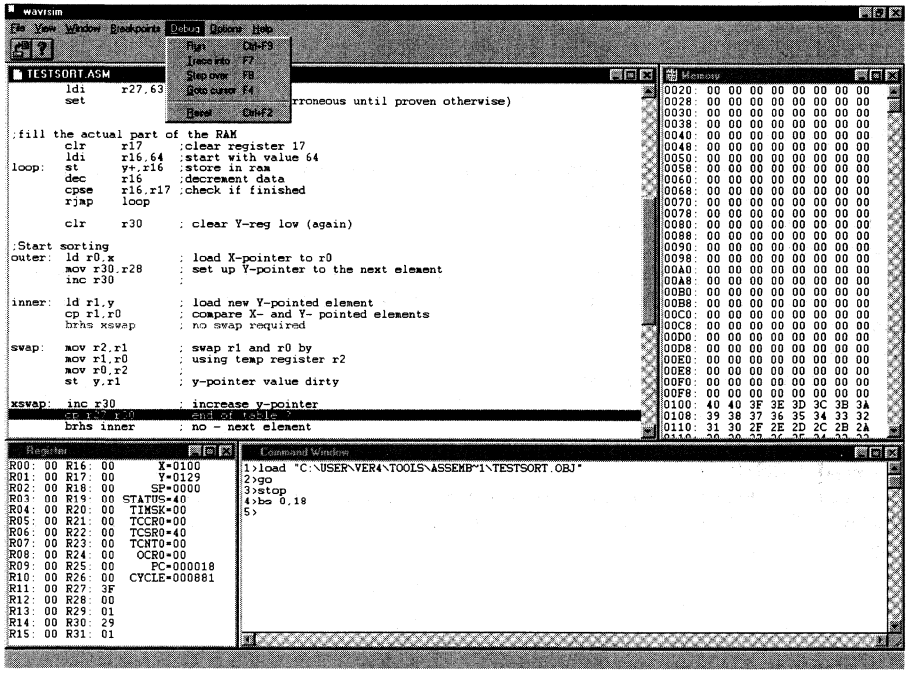
MS Windows Application

The assembler executes under the Windows environment. The Windows version can be executed under Windows 3.11, Windows 95 and Windows NT. The Windows version includes a full editor for writing assembly programs. An MS-DOS command line version is also available.

Atmel *AVR* Simulator

Features:

- Supports the whole range of AT90S microcontrollers
- Assembly source level simulation
- Powerful debugging facilities
- Full support of *AVR* peripheral devices
- Easy to use MS-Windows interface



Debugging Facilities

The simulator has a number of functions to help the programmer to debug programs including:

- Breakpoints: Set up to 256 breakpoints in the source window, and program execution will halt upon reaching one of the breakpoints.
- Single stepping: Step through the code instruction by instruction and watch the execution.
- Step into/Trace over: Select whether calls should be traced, or if these simulation details should be omitted.
- Goto cursor: Place the cursor on an instruction, and the simulator will execute until the marked instruction is reached.
- Run from file. The user can write scripts consisting of simulator commands.

- Display of registers and memory. The user can view all memory spaces, all general purpose working registers and the registers in the I/O map. The user also has the ability to write values in these memories and registers.
- Logging. All information, including register contents, memory contents and I/O accesses can be logged for each instruction.
- The simulator holds control on the number of clock cycles elapsed.

All commands are available through a command window and through menus.

Simulation of Peripherals

The simulator supports the peripheral devices of the *AVR* microcontrollers, including:

- Interrupts. Each interrupt can be set in each cycle enabling complex combinations including nested interrupts.
- Timer/Counters. The timer/counters can also be simulated. This of course includes generating interrupts on overflows and compare matches. Free running mode is also supported.
- I/O ports. The I/O ports are implemented, giving the user the ability to communicate with the simulated programs through the ports.

Together with the powerful control mechanisms present, this makes the simulator a complete debugging tool for the *AVR* family of Enhanced RISC Microcontrollers.

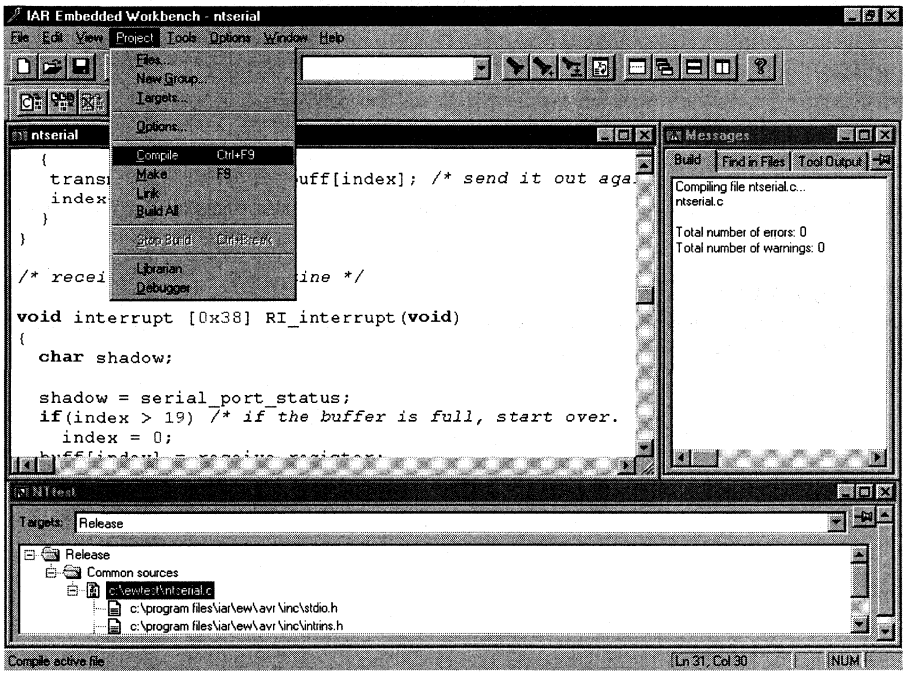
MS Windows Application

The simulator is developed for execution under the Windows environment. The Windows version can be executed under Windows 3.11, Windows 95 and Windows NT.

IAR Development Tools for Atmel *AVR*

Features:

- Fully ANSI Compatible C Compiler
- Includes Embedded Workbench
- C and Assembler level language debugger
- Runs under DOS, Windows 3.11, Windows 95 and Windows NT



Embedded Workbench

The Embedded Workbench offers a total integration of C compiler, editor, linker, librarian, assembler, editor and debugger in a seamless environment. The Embedded Workbench includes the following features:

- Flexible editor. The editor offers flexibility in terms of customizable toolbar and user defined shortcuts. The editor implements the basic Windows editing commands as well as extensions for C programming, such as C syntax coloring and direct jump to context from error listing.
- The hierarchical project maintenance facility makes it possible to have several targets, such as a release target and a debug target with different option settings. Each target is built from one or several groups which in turn are built from one or several files. Compiler options can be set on each level.

- The Make system automatically generates a dependency list of output files, source files and include files. This allows the Make system to only compile, recompile or reassemble the updated parts of the source code, which speeds up the building process.
- Extensive on-line help function makes it easy to quickly find specific help about the tool without leaving the Embedded Workbench.
- The compiler is also available under MS-DOS in a mouse controlled menu driven user interface, allowing all development steps to be performed in an integrated DOS environment.

C Compiler

The C Compiler is an optimizing ANSI C compatible C compiler. The C compiler is the core product in the Embedded Workbench. All data types required by ANSI are supported. Full ANSI C compatibility also means that the compiler conforms to all requirements placed by ANSI on run-time behaviour. The C Compiler includes the following features:

- Fully compatible with the ANSI C standard
- Fully reentrant code
- Absolute read/write of I/O locations at C level
- Built-in AT90S specific Optimizer
- AT90S specific extensions
- Full floating point support
- Four different memory models to allow best fit selection

Assembler

The C compiler kit comes with a relocatable structured assembler. This provides the option of coding time critical sections of the application in assembly without loosing the advantages of the C language. The preprocessor of the C language is incorporated in the assembler, thus allowing use of the full ANSI C macro language, with conditional assembly, macro definitions, if statements and more. C include files can also be used in an assembly program. All modules written in assembly can easily be accessed from C and vice-versa, making the interface between C and assembly a straightforward process.

Linker

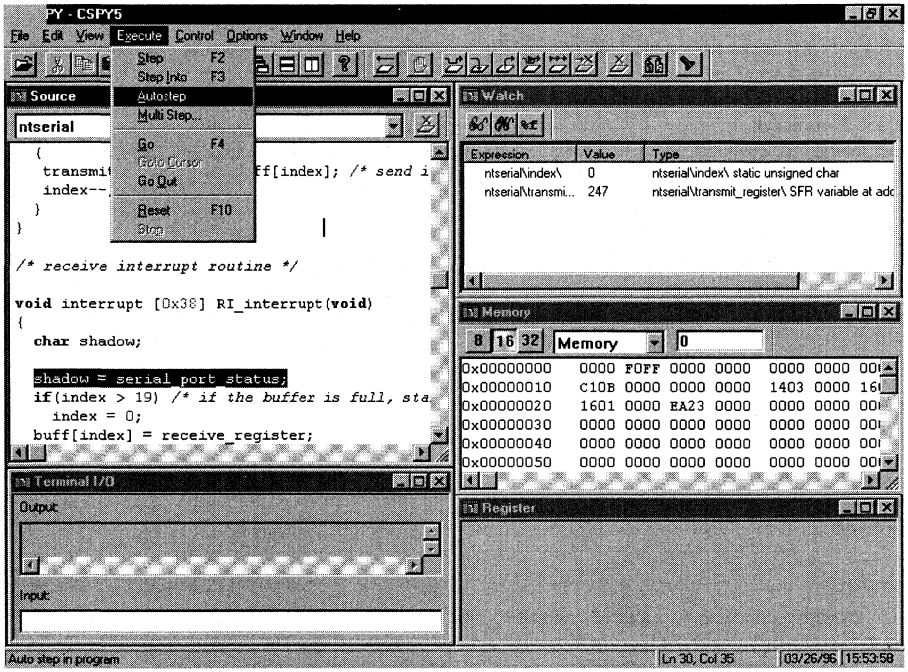
The linker XLINK supports complete linking, relocation and format generation to produce AT90S code. Over 30 output formats are supported. Detailed cross reference and map listing with segments, symbol information, variable locations and function addresses are easily generated.

Librarian

The librarian XLIB creates and maintains libraries and library modules. Listings of modules, entry points and symbolic information contained in every library are easily generated. XLIB can also give the library attribute for conditional loading or the program attribute for unconditional loading.

C-SPY High Level Language Debugger and Simulator

The C-SPY high level language debugger/simulator combines the detailed control code execution needed for embedded development debugging with the flexibility and power of the C language. The source window can display C source and mix it with assembly source. No extra hardware is needed since instruction execution is simulated.



C-SPY includes the following features:

- Powerful breakpoint setting. C-SPY has an unlimited number of breakpoints. Breakpoints can be set on C statements, assembler instructions, and on any address with access type of read, write and opcode fetch. It may also be set as a combination of these. After triggering a breakpoint, a macro command can be executed.
- C-like macro language. A powerful C-like macro language can tailor the environment used for debugging in the C-SPY, including system macros for host file I/O simulation, reset, start up and shut down, as well as statements such as for loops, while loops, if and return statements.
- Interrupt simulation implements commands to launch specific interrupts at a specific cycle count or periodically.
- I/O simulation. C-SPY terminal I/O emulation offers a console window for target system I/O. This unique feature is useful for debugging embedded applications when logical flows are of interest or the target is not yet ready.
- The watch points window makes it possible to watch any expression. The window itself will be updated whenever a breakpoint is trigged or a step is finished. Any variable can be modified during the execution by using specific C expressions.
- The source window for the assembler debugger displays the assembler instructions. It has a built-in assembler and disassembler function, menu and register window, and can evaluate assembler expressions.

Atmel *AVR* In-Circuit Emulator

Features:

- Supports the whole range of AT90S microcontrollers
- Full visibility of all MCU resources
- Powerful breakpoint facilities
- Extensive execution control
- Supports 3 download modes
- Real time emulation
- Software adjustable clock speed
- Supports all on-chip peripherals
- Serial- and parallel port interfacing
- Includes simple programmer
- Integrated with other *AVR* development tools
- 2 in 1: Emulator and programmer

Full visibility

Using the emulator, the status of all resources can be monitored, and most of them can also be modified:

- The register file (R/W)
- SRAM (R/W)
- Program memory (R/W)
- EEPROM (R/W)
- Program Counter (R/W)
- I/O locations (R/W)

Powerful breakpoint facilities

The emulator incorporates powerful breakpoint facilities including:

- SRAM address breakpoint: Break when a specified address in the SRAM is read or written.
- SRAM data breakpoint: Break when a specified value is written to or read from SRAM.
- SRAM address and data breakpoint: An SRAM address breakpoint can be combined with an SRAM data breakpoint.
- Program memory address breakpoint: Break when a specified program memory address is accessed.
- Program memory data breakpoint: Break when a specified value is read from the program memory.
- Register match breakpoint: Break when a specified value is written to one of the 32 registers.
- External trigger breakpoint: Break when an external signal is rising or falling.

Most of the breakpoints can be combined with an event counter in order to break when a break condition is met a specified number of times.

Extensive execution control

The emulator features extensive execution control:

- Single step execution: The emulator executes one instruction and then stops.
- Multiple step execution: The emulator executes a specified number of instructions and then stops.
- Software controlled Trace into/Step over.
- Start/Resume/Stop execution.
- Reset emulator.
- Animation.

Miscellaneous

- The 32 registers, the breakpoint event counter and the I/O data ports can also be monitored during real time emulation. The readout frequency is 10Hz.
- 5 trigger outputs are provided for connection to a DSO or a Logic Analyzer.
- Serial- and parallel port interface. The emulator can be connected to the PC through a standard serial- or parallel port.
- Simple programmer. A device present in the socket can be programmed from the PC or from the emulators program memory.
- In-circuit programming capabilities.
- Supports a wide range of download file formats like Intel Hex and Motorola S-Records.
- Program memory is non-volatile.
- Fast download time.

EQUINOX Micro-Pro *AVR* Device Programmer

Features

- Programs the entire range of *AVR* Flash-based microcontrollers
- Field programmable hardware - allows new devices to be added via a simple software update
- Fast device programming speeds due to optimized FPGA hardware for each target device
- Fast data transfer via PC parallel port
- Comprehensive front-end programming software including buffer editor
- Accepts up to 40-pin DIL devices directly without adapter

Compatible with most PCs

- Runs on IBM XT up to Pentium PC
- Compatible with desktop and Laptop PCs
- Connects to a spare PC parallel port via cable supplied
- Compatible with both standard and enhanced parallel ports

Comprehensive menu driven software:

- DOS-based - Microsoft Windows and Windows compatible
- Buffer editor - color coded displays HEX and ASCII values
- Buffer - Blank Check / Erase / View / Fill / Edit / Search / Copy / Goto
- Device - Select / Guess / Check Signature / Orientation / Auto Program / Blank Check / Erase / Program / Read / Verify / Secure
- File - Load / Save / View - Intel HEX & Binary formats
- <Auto-Program> Hot Key - Erase, program, verify & secure with one key-press
- <Load Last> Hot Key - Automatically loads last file into buffer with one key press

Micro-Pro *AVR* Programming System contents:

- Micro-Pro *AVR* device programmer
- PC Software on 3.5" HD diskette
- Power supply
- PC parallel cable
- Atmel Microcontroller Data Book

Optional programmer accessories:

- PLCC-44 to DIL-40 Package adapter (AD-PLCC44-A)
- SOIC-20 to DIL-20 Package Adapter (AD-SOIC20-A)
- In-circuit emulator / re-programming adapter for the *AVR* family

Contents

7

Contents

ıble of Contents

ıch Atmel data sheet includes an Ordering Information Section which specifies the package ɔes available. This section provides size specifications and outlines for all package types.[1]

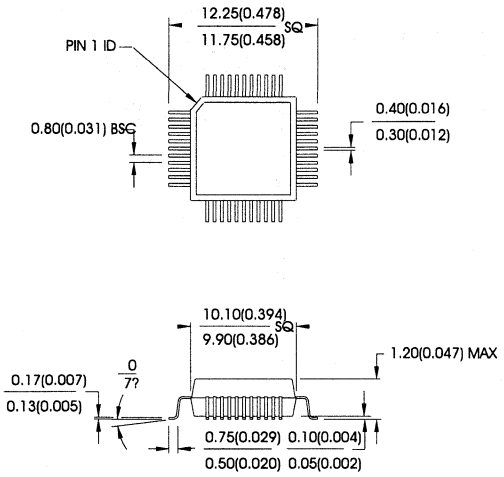
Standard Package Outlines

7

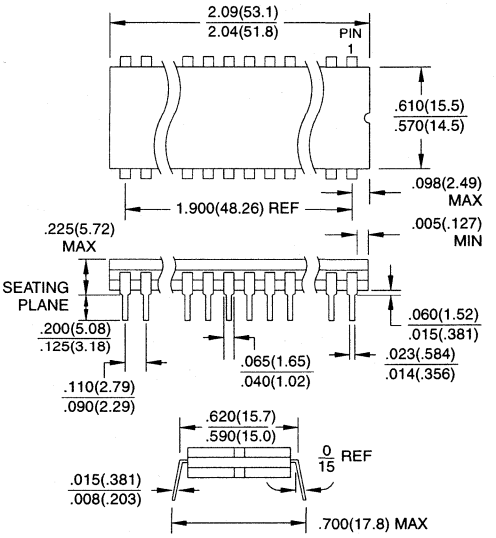
ɔte: 1. Dimensions shown do not include lead plating or mold flash.

0502A

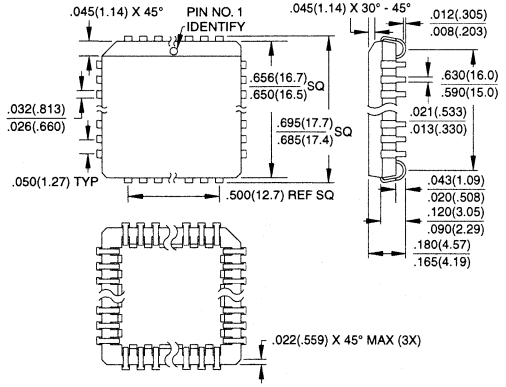
44A, 44 Lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Millimeters and (Inches)*



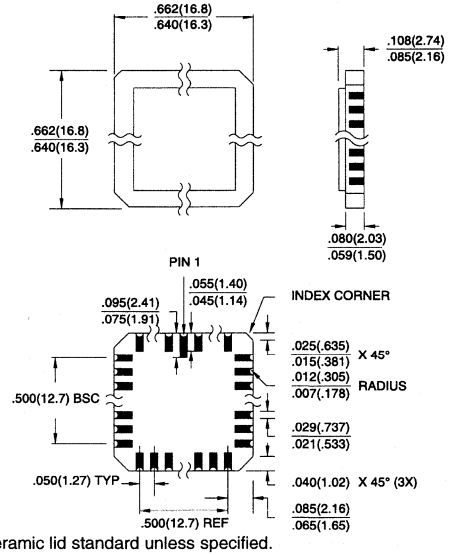
40D6, 40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
Dimensions in Inches and (Millimeters)
MIL-STD-1835 D-5 CONFIG A



44J, 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

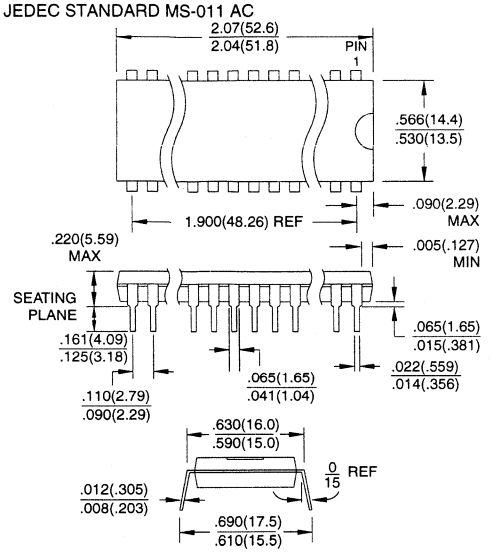


44L, 44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
Dimensions in Inches and (Millimeters)*
MIL-STD-1835 C-5

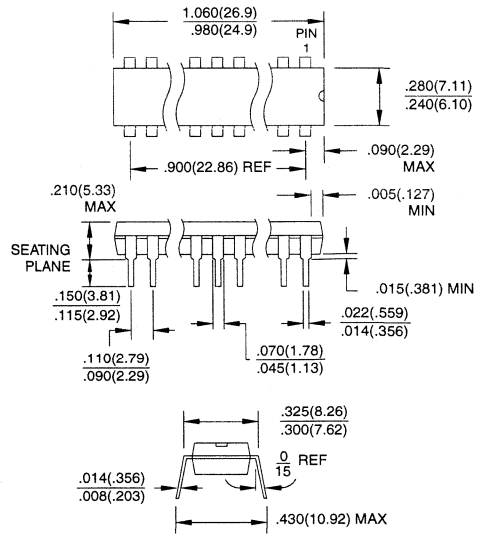


*Ceramic lid standard unless specified.

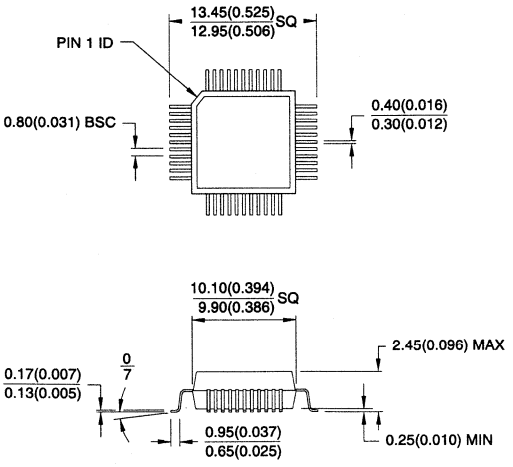
40P6, 40 Lead, 0.600" Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-011 AC



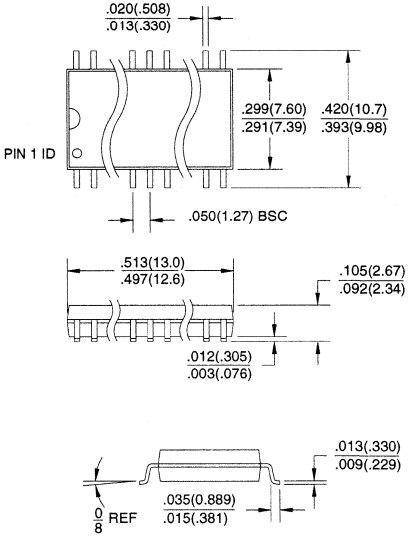
20P3, 20 Lead, 0.300" Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-001 AD



44Q, 44 Lead, Plastic Gull Wing Quad Flat
Package (PQFP)
Dimensions in Inches and (Millimeters)



20S, 20 Lead, 0.300" Wide,
Plastic Gull Wing Small Outline (SOIC)
Dimensions in Inches and (Millimeters)



7

Contents

8

Contents

AVR™ Enhanced RISC Microcontrollers

art Number	Memory Size	Description
T90S1300	1K x 8	***AVR*** Microcontroller with 1K bytes Flash and 128-Bytes EEPROM
T90S2312	2K x 8	***AVR*** Microcontroller with 2K bytes Flash and 128-Bytes EEPROM
T90S8414	8K x 8	***AVR*** Microcontroller with 8K bytes Flash and 256-Bytes EEPROM

51 Family Microcontrollers

art Number	Memory Size	Description
T89C51	4K x 8	80C31 Microcontroller with 4K bytes Flash
T89LV51	4K x 8	2.7-Volt, 80C31 Microcontroller with 4K bytes Flash
T89C52	8K x 8	80C32 Microcontroller with 8K bytes Flash
T89LV52	8K x 8	2.7-Volt, 80C32 Microcontroller with 8K bytes Flash
T89C1051	1K x 8	80C31 Microcontroller with 1K bytes Flash, 20-Pin Package
T89C2051	2K x 8	80C31 Microcontroller with 2K bytes Flash, 20-Pin Package
T89S8252	8K x 8	Downloadable Microcontroller with 8K bytes Flash and 2K bytes E^2PROM
T89C55	20K x 8	80C32 Microcontroller with 20K bytes Flash

EPROMs

Part Number	Organization	Speeds	Description
Battery-Voltage™ (2.7V to 3.6V)			
AT27BV256	32K x 8	70-150 ns	256K bit, 2.7-Volt to 3.6-Volt
AT27BV512	64K x 8	90-150 ns	512K bit, 2.7-Volt to 3.6-Volt
AT27BV010	128K x 8	90-150 ns	1M bit, 2.7-Volt to 3.6-Volt EPROM
AT27BV1024	64K x 16	120-150 ns	1M bit, 2.7-Volt to 3.6-Volt
AT27BV020	256K x 8	120-150 ns	2M bit, 2.7-Volt to 3.6-Volt EPROM
AT27BV4096	256K x 16	150 ns	4M bit, 2.7-Volt to 3.6-Volt
AT27BV040	512K x 8	150 ns	4M bit, 2.7-Volt to 3.6-Volt EPROM
Low Voltage (3.0 to 3.6V)			
AT27LV256A	32K x 8	70-150 ns	256K bit, 3-Volt EPROM
AT27LV512A	64K x 8	90-150 ns	512K bit, 3-Volt EPROM
AT27LV010A	128K x 8	90-150 ns	1M bit, 3-Volt EPROM
AT27LV020A	256K x 8	120-150 ns	2M bit, 3-Volt EPROM
AT27LV040A	512K x 8	150 ns	4M bit, 3-Volt EPROM
Standard Voltage (5V)			
AT27C256R	32K x 8	45-150 ns	256K, 5-Volt EPROM
AT27C512R	64K x 8	45-150 ns	512K, 5-Volt EPROM
AT27C516	32K x 16	45-150 ns	512K, 5-Volt EPROM
AT27C1024	64K x 16	45-150 ns	1M bit, 5-Volt EPROM
AT27C010,L	128K x 8	45-150 ns	1M bit, 5-Volt EPROM, Standard & Low Power
AT27C2048	128K x 16	70-150 ns	2M bit, 5-Volt EPROM
AT27C020	256K x 8	85-150 ns	2M bit, 5-Volt EPROM
AT27C4096	256K x 16	85-150 ns	4M bit, 5-Volt EPROM
AT27C040	512K x 8	80-150 ns	4M bit, 5-Volt EPROM
AT27C080	1024K x 8	100-150 ns	8M bit, 5-Volt EPROM

E^2 Logic Family: E^2PROM + Gate Array

Device Name	E^2PROM Bits	Number of Usable Gates	Number of I/O	Package Type
AT88SC200	2,048	800	8	PDIP/SOIC
AT88SC210	2,048	1,000	64	PQFP/TQFP/PLCC
AT88SC230	2,048	3,300	100	PQFP/TQFP/PLCC
AT88SC2100	2,048	10,000	48	PQFP/TQFP/PLCC
AT88SC410	4,096	1,300	24	PDIP/SOIC
AT88SC1610	16,384	1,500	8	PDIP/SOIC
AT88SC16350	16,384	35,000	144	PQFP/TQFP

lash

art Number	Organization	Speeds	Description
attery-Voltage™ (2.7V to 3.6V)			
T29BV010A	128K x 8	200-350 ns	1M bit, 2.7-Volt Read and 2.7-Volt Write Flash
T29BV020	256K x 8	250-350 ns	2M bit, 2.7-Volt Read and 2.7-Volt Write Flash
T29BV040A	512K x 8	250-350 ns	4M bit, 2.7-Volt Read and 2.7-Volt Write Flash
ow Voltage (3V to 3.6V)			
T29LV256	32K x 8	150-250 ns	256K, 3-Volt Read and 3-Volt Write Flash
T29LV512	64K x 8	150-250 ns	512K, 3-Volt Read and 3-Volt Write Flash
T29LV010A	128K x 8	150-250 ns	1M bit, 3-Volt Read and 3-Volt Write Flash
T29LV1024	64K x 16	150-250 ns	1M bit, 3-Volt Read and 3-Volt Write Flash
T29LV020	256K x 8	200-250 ns	2M bit, 3-Volt Read and 3-Volt Write Flash
T29LV040A	512K x 8	200-250 ns	4M bit, 3-Volt Read and 3-Volt Write Flash
tandard Voltage (5V)			
T29C256	32K x 8	70-250 ns	256K, 5-Volt Read and 5-Volt Write Flash
T29C257	32K x 8	70-250 ns	256K, 5-Volt Read and 5-Volt Write Flash
T29C512	64K x 8	70-200 ns	512K, 5-Volt Read and 5-Volt Write Flash
T29C1024	64K x 16	70-200 ns	1M bit, 5-Volt Read and 5-Volt Write Flash
T29C010A	128K x 8	70-200 ns	1M bit, 5-Volt Read and 5-Volt Write Flash
T29C020	256K x 8	100-200 ns	2M bit, 5-Volt Read and 5-Volt Write Flash
T29C040A	512K x 8	120-250 ns	4M bit, 5-Volt Read and 5-Volt Write Flash
T49F2048	128K x 16	90-200 ns	2M bit, 5-Volt Read and 5-Volt Write Flash
T49F4096	256K x 16	90-200 ns	4M bit, 5-Volt Read and 5-Volt Write Flash
T49F080	1M x 8	120-200 ns	8M bit, 5-Volt Read and 5-Volt Write Flash
T49F8192	512K x 16	120-200 ns	8M bit, 5-Volt Read and 5-Volt Write Flash

lash Memory Cards

art Number	Organization	V_{CC}	Description
T5FC001	1M byte	5-Volt	PCMCIA Compatible Flash Memory Card
T5FC002	2M byte	5-Volt	PCMCIA Compatible Flash Memory Card
T5FC004	4M byte	5-Volt	PCMCIA Compatible Flash Memory Card
T5FC008	8M byte	5-Volt	PCMCIA Compatible Flash Memory Card

8

(Cache Logic™ FPGAs)

Part Number	Registers	Usable Gates	Frequency	Description
AT6002	1.024	2K-4K	250 MHz	96 I/O Pins, 5-Volt, Very Low Power
AT6003	1,600	3K-6K	250 MHz	120 I/O Pins, 5-Volt, Very Low Power
AT6005	3,136	5K-10K	250 MHz	108 I/O Pins, 5-Volt, Very Low Power
AT6010	6,400	10K-20K	250 MHz	204 I/O Pins, 5-Volt, Very Low Power
Low Voltage				
AT6002LV	1.024	2K-4K	250 MHz	96 I/O Pins, 3-Volt, Very Low Power
AT6003LV	1,600	3K-6K	250 MHz	120 I/O Pins, 3-Volt, Very Low Power
AT6005LV	3,136	5K-10K	250 MHz	108 I/O Pins, 3-Volt, Very Low Power
AT6010LV	6,400	10K-20K	250 MHz	204 I/O Pins, 3-Volt, Very Low Power

FPGA Design Development Software

FPGA design tools are available across a broad range of CAE tool vendors and PC and workstation platforms. Design methods supporte include: schematic capture, logic synthesis (VHDL and Verilog), PLD entry (ABEL and CUPL), and automatic component generation of har macros for user-parametrized structured logic (arithmetic elements, counters, registers, encoders, decoders, and other common functions) Refer to current Configurable Logic Data Book.

CAE Tool Support:

Cadence, Exemplar, Intergraph, Mentor, Synopsys, ViewLogic.

Platform Support:

PC, SUN Workstations, HP Workstations.

FPGA Serial Configuration E²PROM

Part Number	Memory Size	Description
AT17C65	65,536 x 1	65K FPGA Configuration E²PROM
AT17C128	131,072 x 1	128K FPGA Configuration E²PROM
AT17C256	262,144 x 1	256K FPGA Configuration E²PROM

Gate Arrays/Embedded Arrays

Part Number	Gates	Description
ATL50 Series	4K-1120K	0.5-Micron CMOS Gate Array, 2.0-Volt & 3.3-Volt Mixed Voltage Operation, 16 Versions with Various Pin & Gate Counts, Memory, Megacells
ATLS60 Series	12.5K-150K	0.6-Micron CMOS Gate Array, 3.3-Volt & 5.0-Volt Operation, Staggered Row Bond Pads, 8 Versions with Various Pin & Gate Counts, Memory, Megacells
ATL60 Series	4K-1120K	0.6-Micron CMOS Gate Array, 3.3-Volt & 5.0-Volt Operation, 16 Versions with Various Pin & Gate Counts, Memory, Megacells
ATL80 Series	2K-600K	0.8-Micron CMOS Gate Array, 3.3-Volt & 5.0-Volt Operation, 12 Versions with Various Pin & Gate Counts, Memory, Megacells
ATLV Series	2K-35K	1.0-Micron CMOS Gate Array, 1.0-Volt & 3.3-Volt Operation, 8 Underlayers with Various Pin & Gate Counts, Memory, Megacells

Cell-Based ICs

Part Number	Description
ECDM05	0.5-Micron CMOS Cell-Based IC Family, 3.3-Volt Operation, Digital, Analog, Memory, Megacells
ECLP07	0.7-Micron CMOS Cell-Based IC Family, 3.3-Volt Operation, Digital, Analog, Memory, Megacells
ECPD07	0.7-Micron CMOS Cell-Based IC Family, 5.0-Volt Operation, Digital, Analog, Memory, Megacells
ECPD10	1.0-Micron CMOS Cell-Based IC Family, 5.0-Volt Operation, Digital, Analog, Memory, Megacells

arallel E²PROMs

art Number	Organization	Speeds	Description
igh Speed			
T28HC64B	8K x 8	55-120 ns	64K E²PROM with 64-Byte Page & Software Data Protection
T28HC256	32K x 8	70-120 ns	256K E²PROM with 64-Byte Page & Software Data Protection
T28HC256E	32K x 8	70-120 ns	256K E²PROM with Extended Endurance, Standard & Low Power
attery-Voltage™ (2.7V to 3.6V)			
T28BV16	2K x 8	250-300 ns	16K E²PROM, 2.7-Volt
T28BV64	8K x 8	300 ns	64K E²PROM, 2.7-Volt
ow Voltage (3.0V to 3.6V)			
T28LV64B	8K x 8	200-300 ns	64K E²PROM with 64-Byte Page & Software Data Protection, 3.0-Volt
T28LV256	32K x 8	200-300 ns	256K E²PROM with 64-Byte Page & Software Data Protection, 3.0-Volt
T28LV010	128K x 8	200-250 ns	1M bit E²PROM with 128-Byte Page & Software Data Protection, 3.0-Volt
tandard Voltage (5V)			
T28C16	2K x 8	150-250 ns	16K E²PROM
T28C16E	2K x 8	150-250 ns	16K E²PROM with Extended Endurance & Fast Write
T28C17	2K x 8	150-250 ns	16K E²PROM with Ready/$\overline{\text{Busy}}$
T28C17E	2K x 8	150-250 ns	16K E²PROM with Ready/$\overline{\text{Busy}}$ & Extended Endurance & Fast Write
T28C64	8K x 8	120-350 ns	64K E²PROM
T28C64E	8K x 8	120-350 ns	64K E²PROM with Extended Endurance & Fast Write
T28C64X	8K x 8	150-450 ns	64K E²PROM without Ready-$\overline{\text{Busy}}$
T28C64B	8K x 8	150-250 ns	64K E²PROM with 64-Byte Page & Software Data Protection
T28C256	32K x 8	150-350 ns	256K E²PROM with 64-Byte Page & Software Data Protection
T28C256E	32K x 8	150-350 ns	256K E²PROM with Extended Endurance
T28C010	128K x 8	120-250 ns	1M bit E²PROM with 128-Byte Page & Software Data Protection
T28C010E	128K x 8	120-250 ns	1M bit E²PROM with 128-Byte Page & Extended Endurance & Software Data Protection
T28C040	512K x 8	150-250 ns	4M bit E²PROM with 256-Byte Page & Software Data Protection

Programmable Logic Devices

Part Number	Packages	Speeds	Description
Flash-Based			
ATF16V8B	20-Pin	7.5-25 ns	8 FFs, 8 I/O Pins, Standard Power
ATF16V8BQ,BQL	20-Pin	10-25 ns	8 FFs, 8 I/O Pins, Quarter Power, Low Power
ATF16V8C	20-Pin	5-7.5 ns	8 FFs, 8 I/O Pins, Standard Power
ATF16V8CZ	20-Pin	10-15 ns	8 FFs, 8 I/O Pins, Zero Power
ATF20V8B	24-Pin, 28-Pin	7.5-25 ns	8 FFs, 8 I/O Pins, Standard Power
ATF20V8BQ,BQL	24-Pin, 28-Pin	10-25 ns	8 FFs, 8 I/O Pins, Quarter Power, Low Power
ATF22V10B	24-Pin, 28-Pin	7.5-25 ns	10 FFs, 10 I/O Pins, Standard Power
ATF22V10BQ,BQL	24-Pin, 28-Pin	15-25 ns	10 FFs, 10 I/O Pins, Quarter Power, Low Power
ATF22V10C	24-Pin, 28-Pin	5-7.5 ns	10 FFs, 10 I/O Pins, Standard Power
ATF22V10CZ	24-Pin, 28-Pin	10-15 ns	10 FFs, 10 I/O Pins, Zero Power
ATF1500,L	44-Pin	7.5-25 ns	32 FFs, 32 I/O Pins, Standard Power & Low Power
Low Voltage			
ATF16LV8C	20-Pin	10-15 ns	8 FFs, 8 I/O Pins, Low Voltage
ATF16LV8CZ	20-Pin	15-25 ns	8 FFs, 8 I/O Pins, Low Voltage, Zero Power
AT22LV10,L	24-Pin, 28-Pin	20-30 ns	10 FFs, 10 I/O Pins, Standard & Low Power
ATLV750B,BL	24-Pin, 28-Pin	10-15 ns	20 FFs, 10 I/O Pins, Standard & Low Power
ATF1500LV,L	44-Pin	12-25 ns	32 FFs, 32 I/O Pins, Standard & Low Power
5-Volt, EPROM-Based			
AT22V10,L	24-Pin, 28-Pin	15-25 ns	10 FFs, 10 I/O Pins, Standard & Low Power
AT22V10B	24-Pin, 28-Pin	7.5-10 ns	10 FFs, 10 I/O Pins, Standard Power
ATV750,L	24-Pin, 28-Pin	20-25 ns	20 FFs, 10 I/O Pins, Standard & Low Power
ATV750B,BL	24-Pin, 28-Pin	7.5-25 ns	20 FFs, 10 I/O Pins, Standard & Low Power
ATV2500H,L	40-Pin, 44-Pin	25-35 ns	48 FFs, 24 I/O Pins, Standard & Low Power
ATV2500B,BL	44-Pin	12-20 ns	48 FFs, 24 I/O Pins, Standard & Low Power
ATV2500BQ,BQL	40-Pin, 44-Pin	20-25 ns	48 FFs, 24 I/O Pins, Quarter Power, Low Power

Programmable Logic Device Design Software

Part Number	Description
ATDS1100PC	Atmel - Synario Entry
ATDS1120PC	Atmel - Synario Verilog Simulation
ATDS1130PC	Atmel - Synario VHDL Synthesis
ATDS1110PC	Atmel - ABEL (Windows Version)
ATDS1210PC	Atmel - ABEL (DOS Version)
ATDS1320PC	Atmel - ProPLD ViewLogic System

Secure Memory ICs

Part Number	Memory Size	Description
AT88SC101	1024 x 1	1K Serial E^2PROM with Security, 1 Memory Zone, 1024 Bits
AT88SC102	1024 x 1	1K Serial E^2PROM with Security, 2 Memory Zones, 512 Bits Each
AT88SC103	1536 x 1	1K Serial E^2PROM with Security, 3 Memory Zones, 512 Bits Each
AT88SC1601	15,872 x 1	16K Serial E^2PROM with Security, 1 Memory Zone, 15,872 Bits
RF ID ASICs	Up to 16K x 1	Analog, Digital & Memory on Single-Chip ASIC

ierial E²PROMs

Part Number	Organization	Vcc	Description
AT24C01	128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 2-Wire Bus Serial E²PROM, Non-Cascadable
AT24C21	128 x 8	2.5 V	1K, 2-Wire Bus Serial E²PROM, Dual Mode, Plug & Play Operation
AT24C01A	128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 2-Wire Bus Serial E²PROM
AT24C02	256 x 8	1.8, 2.5, 2.7, 5.0 V	2K, 2-Wire Bus Serial E²PROM
AT24C04	512 x 8	1.8, 2.5, 2.7, 5.0 V	4K, 2-Wire Bus Serial E²PROM
AT24C08	1024 x 8	1.8, 2.5, 2.7, 5.0 V	8K, 2-Wire Bus Serial E²PROM
AT24C16	2048 x 8	1.8, 2.5, 2.7, 5.0 V	16K, 2-Wire Bus Serial E²PROM
AT24C164	2048 x 8	1.8, 2.5, 2.7, 5.0 V	16K, 2-Wire Bus Serial E²PROM with Cascadable Feature
AT24C32	4096 x 8	1.8, 2.5, 2.7, 5.0 V	32K, 2-Wire Bus Serial E²PROM with Cascadable Feature
AT24C64	8192 x 8	1.8, 2.5, 2.7, 5.0 V	64K, 2-Wire Bus Serial E²PROM with Cascadable Feature
AT25010	128 x 8	1.8, 2.7, 5.0 V	1K, SPI Bus Serial E²PROM, Supports SPI Mode 0 and 3
AT25020	256 x 8	1.8, 2.7, 5.0 V	2K, SPI Bus Serial E²PROM, Supports SPI Mode 0 and 3
AT25040	512 x 8	1.8, 2.7, 5.0 V	4K, SPI Bus Serial E²PROM, Supports SPI Mode 0 and 3
AT93C46	64 x 16 / 128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 3-Wire Bus Serial E²PROM
AT93C46A	64 x 16	1.8, 2.5, 2.7, 5.0 V	1K, 3-Wire Bus Serial E²PROM
AT93C56	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 3-Wire Bus Serial E²PROM
AT93C57	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 3-Wire Bus Serial E²PROM with Special Address
AT93C66	256 x 16 / 512 x 8	2.5, 2.7, 5.0 V	4K, 3-Wire Bus Serial E²PROM
AT59C11	64 x 16 / 128 x 8	2.5, 2.7, 5.0 V	1K, 4-Wire Bus Serial E²PROM
AT59C22	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 4-Wire Bus Serial E²PROM
AT59C13	256 x 16 / 512 x 8	2.5, 2.7, 5.0 V	4K, 4-Wire Bus Serial E²PROM

orth American Sales ffices

ORTHWEST

325 Orchard Parkway
an Jose, CA 95131
EL (408) 436-4270
AX (408) 436-4314

ORTHEAST

00 Granite Street, #409
raintree, MA 02184
EL (617) 849-0220
AX (617) 848-0012

35 Michael Cowpland Dr., #203
anata, Ontario K2M 2E9
anada
EL (613) 599-5338
AX (613) 599-5337

MID-ATLANTIC

01 Carnegie Center, #219
rinceton, NJ 08540
EL (609) 520-0606
AX (609) 520-9175

SOUTHEAST

309 Spring Forest Road, #600
Raleigh, NC 27609
TEL (919) 850-9889
FAX (919) 850-9894

NORTH CENTRAL & EAST CENTRAL

1721 Moon Lake Blvd., #430
Hoffman Estates, IL 60194
TEL (847) 310-1200
FAX (847) 310-1650

SOUTH CENTRAL

11782 Jollyville Rd.
Austin, TX 78759
TEL(512) 219-4050
FAX (512) 219-4051

17304 Preston Road, Suite 720
Dallas, TX 75252
TEL (214) 733-3366
FAX (214) 733-3163

SOUTHWEST

8101 Kaiser, Suite 140
Anaheim Hills, CA 92808
TEL (714) 282-8080
FAX (714) 282-0500

International Sales Offices

AUSTRIA

Atmel GmbH
Niederlassung Oesterreich
Untere Rentmeistergasse 12
3170 Hainfeld
Austria
TEL (43) 2764-3555
FAX (43) 2764-3538

DENMARK

Naverland 2
2600 Glostrup
Denmark
TEL (45) 4343-0801
FAX (45) 4343-0861

FINLAND

Atmel OY
Sinikalliontie 5
02630 Espoo
Finland
TEL (358) 0-5023026
FAX (358) 0-5023126

FRANCE

Atmel Southern Europe
55 Avenue Diderot
94100 St. Maur Des Fosses
Paris, France
TEL (33) 1-48855522
FAX (33) 1-48855596

GERMANY

Atmel GmbH
Ginnheimer Strasse 45
D-60487 Frankfurt 90
Germany
TEL (49) 69-7075910
FAX (49) 69-7075912

Atmel GmbH
Niederlassung Sud
Litzdorfer Strasse 11
D-83064 Raubling
Germany
TEL (49) 8034-9127
FAX (49) 8034-9330

HONG KONG

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road, Tsimshatsui East
Kowloon
Hong Kong
TEL (852) 27219778
FAX (852) 27221369

ITALY

Ufficio di Milano
Centro Direzionale Colleoni
Palazzo Andromeda 3
20041 Agrate Brianza
Italy
TEL (39) 39 605 69 55
FAX (39) 39 605 69 69

JAPAN

Atmel Japan K.K.
Thomas Bldg., 16-1
Nihonbashi Hakosaki-Cho
Chuo-Ku, Tokyo 103
Japan
TEL (81) 3-5641-0211
FAX (81) 3-5641-0217

KOREA

Atmel Korea, Ltd.
6F, Norsan Bldg., 106-8
Guro 5—Dong, Guro-Ku
Seoul, Korea (152-055)
TEL (82) 2-8396341
FAX (82) 2-8396343

SINGAPORE

Atmel Singapore PTE., Ltd.
6001 Beach Road
Golden Mile Tower #21-01
Singapore 0719
TEL (65) 2999-212
FAX (65) 2910-955

SWEDEN

Atmel Sweden
P.O. Box 142
S-19422 Upplands Vasby
Sweden
TEL (46) 8-590-74910
FAX (46) 8-590-74940

TAIWAN

Atmel Taiwan Ltd.
FL 15-4, No. 83, Sec. 1
Nan-Kan Road
Lu Chu Hsiang, Taoyuan Hsien
Taiwan, R.O.C.
TEL (886) 3-3229133
FAX (886) 3-3229131

UNITED KINGDOM

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686677
FAX (44) 1276-686697

0411D

Atmel Operations

ATMEL COLORADO SPRINGS

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

ATMEL ROUSSET

Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 42 53 60 00
FAX (33) 42 53 60 01

Atmel Development Centers

ATMEL CHESAPEAKE

6740 Alenader Bell Dr., #100
Columbia, MD 21046
TEL (410) 312-4400
FAX (410) 312-4424

ATMEL NORWAY

Vestre Rosten 78
7075 Tiller
Norway
TEL (47) 72 88 87 20
FAX (47) 72 88 87 18

Alabama

ALL AMERICAN SEMICONDUCTOR
950 Corporate Dr., Suite 115D
Huntsville, AL 35805
TEL (205) 837-1555
FAX (205) 837-7733

ARROW/SCHWEBER ELECTRONICS
015 Henderson Road
Hunstville, AL 35816
TEL (205) 837-6955
FAX (205) 721-1581

INSIGHT ELECTRONICS, INC.
4835 University Square,
Suite 19
Huntsville, AL 35816
TEL (205) 830-1222
FAX (205) 830-1225

MARSHALL INDUSTRIES
3313 Memorial Parkway South
Huntsville, AL 35801
TEL (205) 881-9235
FAX (205) 881-1490

MILGRAY/HUNTSVILLE
5021 Bradford Drive
Suite 202
Huntsville, AL 35805
TEL (205) 722-9709
FAX (205) 722-0161

PIONEER TECHNOLOGIES
4835 University Square,
Suite 5
Huntsville, AL 35816
TEL (205) 837-9300
FAX (205) 837-9358

Arizona

ARROW/SCHWEBER ELECTRONICS
2415 West Erie Drive
Tempe, AZ 85282
TEL (602) 431-0030
FAX (602) 431-9555

INSIGHT ELECTRONICS, INC.
1515 West University,
Suite 103
Tempe, AZ 85281
TEL (602) 829-1800
FAX (602) 967-2658

MARSHALL INDUSTRIES
9831 S. 51st Street,
Suite C107-109
Phoenix, AZ 85044
TEL (602) 496-0290
FAX (602) 893-9029

PIONEER STANDARD ELECTRONICS
1438 West Broadway,
Suite B-140
Tempe, AZ 85282
TEL (602) 350-9335
FAX (602) 350-9376

California

ALL AMERICAN SEMICONDUCTOR
230 Devcon Drive
San Jose, CA 95112
TEL (408) 441-1300
FAX (408) 437-8970

26010 Mureau Road, Suite 120
Calabasas, CA 91302
TEL (818) 878-0555
FAX (818) 878-0533

10805 Holder Street, Suite 100
Cypress, CA 90630
TEL (714) 229-8600
FAX (714) 229-8603

6390 Greenwich Drive,
Suite 170
San Diego, CA 92122
TEL (619) 658-0200
FAX (619) 658-0201

111 S. Court Street, Suite 104
Visalia, CA 93291
TEL (209) 734-8861
FAX (209) 734-8865

ARROW/SCHWEBER ELECTRONICS
6 Cromwell St., Suite 100
Irvine, CA 92718
TEL (714) 587-0404
FAX (714) 454-4206

Malibu Canyon Business Park
26677 West Agoura Road
Calabasas, CA 91302
TEL (818) 880-9686
FAX (818) 880-4687

9511 Ridgehaven Court
San Diego, CA 92123
TEL (619) 565-4800
FAX (619) 279-0862

1180 Murphy Avenue
San Jose, CA 95131
TEL (408) 441-9700
FAX (408) 453-4810

INSIGHT ELECTRONICS, INC.
4333 Park Terrace Dr.,
Suite 101
Westlake Village, CA 91361
TEL (818) 707-2101
FAX (818) 707-0321

2 Venture Plaza, Suite 340
Irvine, CA 92718
TEL (714) 727-3291
FAX (714) 727-1804

9980 Huennekens Street
San Diego, CA 92121
TEL (619) 677-3100
FAX (619) 677-3131

1295 Oakmead Parkway
Sunnyvale, CA 94086
TEL (408) 720-9222
FAX (408) 720-8390

JAN DEVICES
6925 Canby Avenue,
Building 109
Reseda, CA 91335
TEL (818) 757-2000
FAX (818) 708-7436

MARSHALL INDUSTRIES
9320 Telstar Avenue
El Monte, CA 91731
TEL (818) 307-6000
FAX (818) 307-6173

2941 Sunrise Blvd., Suite 130
Rancho Cordova, CA 95742
TEL (916) 635-9700
FAX (916) 635-6044

336 Los Coches Street
Milpitas, CA 95035
TEL (408) 942-4600
FAX (408) 262-1224

One Morgan
Irvine, CA 92718
TEL (714) 859-5050
FAX (714) 581-5255

26537 Agoura Road
Calabasas, CA 91302
TEL (818) 878-7000
FAX (818) 880-6846

5961 Kearny Villa
San Diego, CA 92123
TEL (619) 627-4184
FAX (619) 627-4163

MILGRAY ELECTRONICS, INC.
2880 Zanker Road, Suite 102
San Jose, CA 95134
TEL (408) 456-0900
FAX (408) 456-0300

275 E. Hillcrest Dr., Suite 145
Thousand Oaks, CA 91360
TEL (805) 371-9399
FAX (805) 371-9317

MILGRAY/ORANGE COUNTY
25 Mauchly, Suite 329
Irvine, CA 92718
TEL (714) 753-1282
FAX (714) 753-1682

6835 Flanders Drive, Suite 300
San Diego, CA 92121
TEL (619) 457-7545
FAX (619) 457-9750

PIONEER STANDARD ELECTRONICS
5126 Clareton Drive, Suite 160
Agoura Hills, CA 91301
TEL (818) 865-5800
FAX (818) 865-5814

9449 Balboa Avenue, Suite 114
San Diego, CA 92123
TEL (619) 560-1318
FAX (619) 514-7799

217 Technology Drive,
Suite 110
Irvine, CA 92718
TEL (714) 753-5090
FAX (714) 753-5074

PIONEER TECHNOLOGIES
333 River Oaks Pkwy
San Jose, CA 95134
TEL (408) 954-9100
FAX (408) 954-9113

ZEUS ELECTRONICS
6 Cromwell St., Suite 100
Irvine, CA 92718
TEL (714) 581-4622
(800) 52-HI-REL
FAX (714) 454-4355

6276 San Ignacio Avenue,
Suite E
San Jose, CA 95119
TEL (408) 629-4789
(800) 52-HI-REL
FAX (408) 629-4792

Colorado

ARROW/SCHWEBER ELECTRONICS
101 Inverness Dr. East,
Suite 120
Englewood, CO 80112
TEL (303) 799-0258
FAX (303) 799-0730

INSIGHT ELECTRONICS, INC.
2 Inverness Drive South,
Suite 102
Englewood, CO 80112
TEL (303) 649-1800
FAX (303) 649-1818

MARSHALL INDUSTRIES
12351 North Grant
Thornton, CO 80241
TEL (303) 451-8383
FAX (303) 457-2899

MILGRAY/COLORADO
5650 D T C Parkway,
Suite 202
Englewood, CO 80111
TEL (303) 721-7702
FAX (303) 721-7803

PIONEER TECHNOLOGIES GROUP
5600 Green Wood Plaza Blvd.,
Suite 200
Englewood, CO 80111
TEL 303-773-8090
FAX 303-773-8194

Connecticut

ALL AMERICAN
100 Mill Plain Rd., Suite 360
Danbury, CT 06811
TEL (203) 791-3818
FAX (516) 434-9394

ARROW/SCHWEBER ELECTRONICS
860 N. Main Street Extension
Wallingford, CT 06492
TEL (203) 265-7741
FAX (203) 265-7988

MARSHALL INDUSTRIES
20 Sterling Drive
Barnes Industrial Park North
P.O. Box 200
Wallingford, CT 06492-0200
TEL (203) 265-3822
FAX (203) 284-9285

MILGRAY/CONNECTICUT
326 West Main Street
Milford, CT 06460
TEL (203) 878-5538
FAX (203) 878-6970

PIONEER STANDARD ELECTRONICS
Two Trap Falls
Shelton, CT 06484
TEL (203) 929-5600
FAX (203) 929-979

Florida

ALL AMERICAN SEMICONDUCTOR
16115 N.W. 52nd Avenue
Miami, FL 33014
TEL (305) 621-8282
FAX (305) 620-7831

14450 46th Street
Shelter 116
Clearwater, FL 34622
TEL (813) 532-9800
FAX (813) 538-5567

1400 East Newport Center
Drive,
Suite 205
Deerfield Beach, FL 33442
TEL (305) 429-2800
FAX (305) 429-0391

ARROW/SCHWEBER ELECTRONICS
400 Fairway Dr.
Deerfield Beach, FL 33441
TEL (305) 429-8200
FAX (305) 428-3991

Idg. D, Suite 3101, 3102, 3103
7 Skyline Dr.
ake Mary, FL 32746
EL (407) 333-9300
AX (407) 333-9320

NSIGHT ELECTRONICS, INC.
00 North Lake Blvd.,
uite 250
ltamonte Springs, FL 32701
EL (407) 834-6310
AX (407) 834-6461

400 Congress Avenue,
uite 1600
oca Raton, FL 33487
EL (407) 997-2540
AX (407) 997-2542

7757 US Hwy. 19 North,
uite 520
learwater, FL 34624
EL (813) 524-8850
AX (813) 532-4252

MARSHALL INDUSTRIES
2700 W. Cypress Creek Road,
uite D114
t. Lauderdale, FL 33309
EL (305) 977-4880
AX (305) 977-4887

380 S. Northlake Boulevard,
uite 1024
ltamonte Springs, FL 32701
EL (407) 767-8585
AX (407) 767-8676

2840 Scherer Drive, Suite 410
St. Petersburg, FL 33716
TEL (813) 573-1399
FAX (813) 573-0069

MILGRAY/FLORIDA
755 Rinehart Road, Suite 100
Lake Mary, FL 32746
TEL (407) 321-2555
FAX (407) 322-4225

PIONEER STANDARD ELECTRONICS
674 S. Military Trail
Deerfield Beach, FL 33442
TEL (305) 428-8877
FAX (305) 481-2950

PIONEER TECHNOLOGIES
337 S. Northlake Boulevard
Suite 1000
Altamonte Springs, FL 32701
TEL (407) 834-9090
FAX (407) 834-0865

ZEUS ELECTRONICS
37 Skyline Drive
Bldg. D, Suite 3101
Lake Mary, FL 32746
TEL (407) 333-3055
(800) 52-HI-REL
FAX (407) 333-9681

Georgia

ALL AMERICAN
6875 Jimmy Carter Blvd.,
Suite 3100
Norcross, GA 30071
TEL (770) 441-7500
FAX (770) 441-3660

ARROW/SCHWEBER ELECTRONICS
4250 E. River Green Parkway
Duluth, GA 30136
TEL (770) 497-1300
FAX(770) 476-1493

INSIGHT ELECTRONICS, INC.
3005 Breckinridge Blvd.,
Suite 210-A
Duluth, GA 30136
TEL (404) 717-8566
FAX (404) 717-8588

MARSHALL INDUSTRIES
5300 Oakbrook Parkway,
Suite 140
Norcross, GA 30093
TEL (404) 923-5750
FAX (404) 923-2743

MILGRAY/ATLANTA
3000 Northwoods Parkway
Suite 115
Norcross, GA 30071
TEL (404) 446-9777
FAX (404) 446-1186

PIONEER STANDARD ELECTRONICS
4250C Rivergreen Parkway
Duluth, GA 30136
TEL (404) 623-1003
FAX (404) 623-0665

Illinois

ALL AMERICAN SEMICONDUCTOR
1930 N. Thoreau, Suite 200
Schaumburg, IL 60173
TEL (847) 303-1995
FAX (847) 303-1996

ARROW/SCHWEBER ELECTRONICS
1140 W. Thorndale Ave.
Itasca, IL 60143
TEL (708) 250-0500
FAX (708) 250-0916

INSIGHT ELECTRONICS, INC.
1365 Wiley Road
Suite 142
Schaumburg, IL 60173
TEL (847) 885-9700
FAX (847) 885-9701

MARSHALL INDUSTRIES
50 East Commerce Drive, Unit 1
Schaumburg, IL 60173
TEL (847) 490-0155
FAX (847) 490-0569

MILGRAY/CHICAGO
Kennedy Corporate Center 1
1530 E. Dundee Road, Suite 31
Palatine, IL 60067-8319
TEL (847) 202-1900
FAX (847) 202-1985

PIONEER TECHNOLOGIES
2171 Executive Drive,
Suite 200
Addison, IL 60101
TEL (708) 495-9680
FAX (708) 495-9831

ZEUS ELECTRONICS
1140 W. Thorndale Avenue
Itasca, IL 60143
TEL (708) 595-9730
(800) 52-HI-REL
FAX (708) 595-9896

Indiana

ARROW/SCHWEBER ELECTRONICS
7108 Lakeview Parkway
West Drive
Indianapolis, IN 46268
TEL (317) 299-2071
FAX (317) 299-2379

MARSHALL INDUSTRIES
6990 Corporate Drive
Indianapolis, IN 46278
TEL (317) 297-0483
FAX (317) 297-2787

MILGRAY/INDIANA
5226 Elmwood Avenue
Indianapolis, IN 46203
TEL (317) 781-9997
FAX (317) 781-6970

PIONEER STANDARD ELECTRONICS
9350 N. Priority Way, W. Drive
Indianapolis, IN 46240
TEL (317) 573-0880
FAX (317) 573-0979

Kansas

ARROW/SCHWEBER ELECTRONICS
9801 Legler Road
Lenexa, KS 66219
TEL (913) 541-9542
FAX (913) 752-2612

INSIGHT ELECTRONICS, INC.
8700 Monrovia, Suite 310
Lenexa, KS 66216
TEL (913) 492-0408
FAX (913) 492-0708

MARSHALL INDUSTRIES
10413 West 84th Terrace
Pine Ridge Business Park
Lenexa, KS 68214
TEL (913) 492-3121
FAX (913) 492-6205

MILGRAY/KANSAS CITY
6400 Glenwood, Suite 313
Overland Park, KS 66202
TEL (913) 236-8800
FAX (913) 384-6825

Maryland

ALL AMERICAN
14636 Rothged Drive
Rockville, MD 20850
TEL (301) 251-1205
FAX (301) 251-8574

ARROW/SCHWEBER ELECTRONICS
9800J Patuxent Woods Drive
Columbia, MD 21046
TEL (301) 596-7800
FAX (301) 596-7821

INSIGHT ELECTRONICS, INC.
6925 Oakland Mills Road,
Suite D
Columbia, MD 21045
TEL (410) 381-3131
FAX (410) 381 3141

MARSHALL INDUSTRIES
9130B Guilford Road
Columbia, MD 21046-1803
TEL (301) 470-2800
FAX (301) 622-0451

MILGRAY/WASHINGTON
6460 Dobbin Road, Suite D
Columbia, MD 21045
TEL (410) 730-6119
FAX (410) 730-8940

PIONEER TECHNOLOGIES
9100 Gaither Road
Gaithersburg, MD 20877
TEL (301) 921-0660
FAX (301) 921-4255

15810 Gaither Road
Gaithersburg, MD 20877
TEL (301) 921-3822
FAX (301) 921-3858

Massachusetts

ALL AMERICAN
19A Crosby Drive
Bedford, MA 01730
TEL (617) 275-8888
FAX (617) 275-1982

ARROW/SCHWEBER ELECTRONICS
25 Upton Drive
Wilmington, MA 01887
TEL (508) 658-0900
FAX (508) 694-1754

INSIGHT ELECTRONICS, INC.
55 Cambridge Street, Suite 301
Burlington, MA 01803
TEL (617) 270-9400
FAX (617) 270-3279

MARSHALL INDUSTRIES
33 Upton Drive
Wilmington, MA 01887
TEL (508) 658-0810
FAX (508) 658-7608

MILGRAY/NEW ENGLAND
Ballardvale Park
187 Ballardvale Street
Wilmington, MA 01887
TEL (508) 657-5900
FAX (508) 658-7989

PIONEER STANDARD ELECTRONICS
44 Hartwell Avenue
Lexington, MA 02173
TEL (617) 861-9200
FAX (617) 863-1547

ZEUS ELECTRONICS
25 Upton Drive
Wilmington, MA 01887
TEL (508) 658-4776
(800) 52-HI-REL
FAX (508) 694-2199

Michigan

ARROW/SCHWEBER ELECTRONICS
44720 Helm Street
Plymouth, MI 48170
TEL (313) 455-0850
FAX (313) 455-6656

INSIGHT ELECTRONICS, INC.
802 E. Grand River, Suite 103
Brighton, MI 48116
TEL (810) 229-7710
FAX (810) 229-6435

MARSHALL INDUSTRIES
31067 Schoolcraft
Livonia, MI 48150
TEL (313) 525-5850
FAX (313) 525-5855

PIONEER STANDARD ELECTRONICS
4467 Byron Center Avenue
Wyoming, MI 49509
TEL (616) 534-6074
FAX (616) 534-3922

44190 Plymouth Oaks Drive
Plymouth, MI 48270
TEL (313) 416-2157
FAX (313) 416-2415

Minnesota

ALL AMERICAN SEMICONDUCTOR
7716 Golden Triangle Drive
Eden Prairie, MN 55344
TEL (612) 944-2151
FAX (612) 944-9803

ARROW/SCHWEBER ELECTRONICS
10100 Viking Drive, Suite 100
Eden Prairie, MN 55344
TEL (612) 941-5280
FAX (612) 941-9405

10120 A West 76th Street
Eden Prairie, MN 55344
TEL (612) 946-4800

INSIGHT ELECTRONICS, INC.
5353 Gamble Drive,
Suite 330
St. Louis Park, MN 55416
TEL (612) 525-9999
FAX (612) 525-9998

MARSHALL INDUSTRIES
14800 28th Avenue, North,
Suite 175
Minneapolis, MN 55447
TEL (612) 559-2211
FAX (612) 559-8321

PIONEER STANDARD ELECTRONICS
7625 Golden Triangle
Eden Prairie, MN 55344
TEL (612) 944-3355
FAX (612) 944-3794

Missouri

ARROW/SCHWEBER ELECTRONICS
2380 Schuetz Road
St. Louis, MO 63146
TEL (314) 567-6888
FAX (314) 567-1164

MARSHALL INDUSTRIES
514 Earthcity Expressway,
Suite 131
Earthcity, MO 63045
TEL (314) 770-1749
FAX (314) 770-1486

PIONEER STANDARD ELECTRONICS
111 West Port Plaza,
Suite 625
St. Louis, MO 63146
TEL (314) 542-3077
FAX (314) 542-3078

New Jersey

ARROW/SCHWEBER ELECTRONICS
43 Route 46 East
Pine Brook, NJ 07058
TEL (201) 227-7880
FAX (201) 227-2064

4 East Stow Road, Unit 11
Marlton, NJ 08053
TEL (609) 596-8000
FAX (609) 596-9632

INSIGHT ELECTRONICS, INC.
115 Rt. 46, Suite F-1000
Mountain Lakes, NJ 07046
TEL (201) 316-6040
FAX (201) 335-1495

2 Eves Drive, Suite 208
Marlton, NJ 08053
TEL (609) 985-5556
FAX (609) 985-5895

MARSHALL INDUSTRIES
101 Fairfield Road
Fairfield, NJ 07004
TEL (201) 882-0320
FAX (201) 882-0095

158 Gaither Drive
Mt. Laurel, NJ 08054
TEL (609) 234-9100
FAX (609) 778-1819

MILGRAY/DELAWARE VALLEY
523 Fellowship Road,
Suite 275
Mt. Laurel, NJ 08054
TEL (609) 778-1300
FAX (609) 778-7669

MILGRAY/NEW JERSEY
3799 Route 46 East,
Suite 303
Parsippany, NJ 07054
TEL (201) 335-1766
FAX (201) 335-2110

PIONEER STANDARD ELECTRONICS
14A Madison Road
Fairfield, NJ 07006
TEL (201) 575-3510
FAX (201) 575-3454

New York

ALL AMERICAN SEMICONDUCTOR
275B Marcus Boulevard
Hauppauge, NY 11788
TEL (516)434-9000
FAX (516) 434-9394

333 Metro park
Rochester, NY 14623
TEL (716) 292-6700
FAX (716) 292-6755

ARROW/SCHWEBER ELECTRONICS
120 Commerce Street
Hauppauge, NY 11788
TEL (516) 231-1000
FAX (516) 231-1072

25 Hub Drive
Melville, NY 11747-3509
TEL (516) 391-1556
FAX (516) 391 1640

3375 Brighton-Henrielts
Townline Road
Rochester, NY 14623
TEL (716) 427-0300
FAX (716) 427-0735

INSIGHT ELECTRONICS, INC.
c/o Airlurn Exec.
80 Orville Drive
Bohemia, NY 11716
TEL (516) 244-1640

MARSHALL INDUSTRIES
100 Marshall Drive
Endicott, NY 13760
TEL (607) 785-2345
FAX (607) 785-5546

1250 Scottsville Road
Rochester, NY 14624
TEL (716) 235-7620
FAX (716) 235-0052

3505 Veterans Memorial
Highway Suite L
Ronkonkoma, NY 11779
TEL (516) 737-9300
FAX (516) 737 9580

MILGRAY/NEW YORK
77 Schmitt Boulevard
Farmingdale, NY 11735
TEL (516) 391-3000
FAX (516) 420-0685

MILGRAY/UPSTATE NEW YORK
One Corporate Place, Suite 200
1170 Pittsford Victor Road
Pittsford, NY 14534
TEL (716) 381-9700
FAX (716) 381-9495

PIONEER STANDARD ELECTRONICS
1249 Upper Front, Suite 201
Binghamton, NY 13901
TEL (607) 722-9300
FAX (607) 722-9562

840 Fairport Park
Fairport, NY 14450
TEL (716) 381-7070
FAX (716) 381-5955

One Penn Plaza #2032
New York, NY 10119
TEL (212) 631-4700
FAX (212) 971-0374

60 Crossways Park West
Woodbury, NY 11797
TEL (516) 921-8700
FAX (516) 921-2143

ZEUS ELECTRONICS
100 Midland Avenue
Port Chester, NY 10573
TEL (914) 937-7400
(800) 52-HI-REL
FAX (914) 937-2553

North Carolina

ARROW/SCHWEBER ELECTRONICS
5240 Greens Dairy Road
Raleigh, NC 27604
TEL (919) 876-3132
FAX (919) 878-9517

INSIGHT ELECTRONICS, INC.
9909 Alden Glen Drive
Charlotte, NC 28269
TEL (704) 549-9750
FAX (704) 549-9751

811 Spring Forest Rd.,
Suite 1000
Raleigh, NC 27609
TEL (800) 677-7716

MARSHALL INDUSTRIES
5224 Greens Dairy Road
Raleigh, NC 27604
TEL (919) 878-9882
FAX (919) 872-2431

MILGRAY/RALEIGH
2925 Huntleigh Drive,
Suite 101
Raleigh, NC 27604
TEL (919) 790-8094
FAX (919) 872-8851

PIONEER TECHNOLOGIES
2200 Gateway Center Blvd.,
Suite 215
Morrisville, NC 27560
TEL (919) 460-1530
FAX (919) 460-1540

Ohio

ARROW/SCHWEBER ELECTRONICS
8200 Washington Village Dr.,
Suite A
Centerville, OH 45458
TEL (513) 435-5563
FAX (513) 435-2049

6573 E Cochran Road
Solon, OH 44139
TEL (216) 248-3990
FAX (216) 248-1106

INSIGHT ELECTRONICS, INC.
9700 Rockside Road,
Suite 105
Valley View, OH 44125
TEL (216) 520-4333
FAX (216) 520-4322

MARSHALL INDUSTRIES
30700 Bainbridge Road, Unit A
Solon, OH 44139
TEL (216) 248-1788
FAX (216) 248-2312

3520 Park Center Drive
Dayton, OH 45414
TEL (513) 898-4480
FAX (513) 898-9363

MILGRAY/CLEVELAND
6155 Rockside Road, Suite 206
Cleveland, OH 44131
TEL (216) 447-1520
FAX (216) 447-1761

PIONEER STANDARD ELECTRONICS
2385 Edison Blvd.
Twinsburg, OH 44087
TEL (216) 487-5500
FAX (216) 487-0256

4800 East 131st Street
Cleveland, OH 44105
TEL (216) 587-3600
FAX (216) 587-3906

4433 Interpoint Boulevard
Dayton, OH 45424
TEL (513) 236-9900
FAX (513) 236-8133

100 Old Wilson Bridge Road,
Suite 105
Worthington, OH 43085
TEL (614) 848-4854
FAX (614) 848-4889

Oklahoma

ARROW/SCHWEBER ELECTRONICS
12111 E. 51st Street, Suite 101
Tulsa, OK 74146
TEL (918) 252-7537
FAX (918) 254-0917

PIONEER STANDARD ELECTRONICS
9717 E. 42nd Street, Suite 105
Tulsa, OK 74146
TEL (918) 665-7840
FAX (918) 665-1891

Oregon

ALMAC/ARROW ELECTRONICS
9500 S.W. Nimbus Ave. Bld. E
Beaverton, OR 97008
TEL (503) 629-8090
FAX (503) 645-0611

ALL AMERICAN/PORTLAND
1815 N.W. 169th Place,
Suite 6025
Beaverton, OR 97006
TEL (503) 531-3333
FAX (503) 531-3695

NSIGHT ELECTRONICS, INC.
)705 S.W. Nimbus Avenue,
Suite 200
Beaverton, OR 97008
TEL (503) 644-3103
FAX (503) 641-4530

MARSHALL INDUSTRIES
)705 S.W. Gemini Drive
Beaverton, OR 97005
TEL (503) 644-5050
FAX (503) 646-8256

MILGRAY/OREGON
3705 S.W. Nimbus Ave.,
Suite 260
Beaverton, OR 97008
TEL (503) 626-4040
FAX (503) 641-0650

PIONEER TECHNOLOGIES GROUP
3905 Southwest Nimbus,
Suite 160
Beaverton, OR 97008
TEL 503-626-7300
FAX 503-626-5300

Pennsylvania

ARROW/SCHWEBER ELECTRONICS
2681 Mosside Blvd., Suite 204
Monroeville, PA 15146
TEL (412) 856-9490
FAX (412) 856-9507

INSIGHT ELECTRONICS, INC.
20530, Route 19, Suite 5
Cranberry Township, PA 16066
TEL (412) 779-0060
FAX (412) 779-0070

PIONEER STANDARD ELECTRONICS
259 Kappa Drive
Pittsburgh, PA 15238
TEL (412) 782-2300
FAX (412) 963-8255

PIONEER TECHNOLOGIES
500 Enterprise Road
Horsham, PA 19044
TEL (215) 674-4000
FAX (215) 674-3107

Texas

ALL AMERICAN SEMICONDUCTOR
13706 Research Blvd., Suite 103
Austin, TX 78750
TEL (512) 335-2280
FAX (512) 335-2282

11210 Steeplecrest, Suite 206
Houston, TX 77065
TEL (713) 955-1993
FAX (713) 955-2215

ALL AMERICAN/DALLAS
1771 International Parkway,
Suite 101
Richardson, TX 75081
TEL (214) 231-5300
FAX (214) 437-0353

AMIGA SALES & MARKETING C/O JAN DEVICES
12342 Hunters Chase Blvd.,
Suite 3127
Austin, TX 78729
TEL (818) 757-2005
FAX (818) 708-7436

ARROW/SCHWEBER ELECTRONICS
Braker Center III, Bldg. M1
11500 Metric Blvd., Suite 160
Austin, TX 78758
TEL (512) 835-4180
FAX (512) 832-9875

3220 Commander Drive
Carrollton, TX 75006
TEL (214) 380-6464
FAX (214) 248-7208

19416 Park Row, Suite 190
Westgate Center, Bldg. B
Houston, TX 77084
TEL (713) 647-6868
FAX (713) 492-8722

INSIGHT ELECTRONICS, INC.
11500 Metric Boulevard,
Suite 215
Austin, TX 78758
TEL (512) 719-3090
FAX (512) 719-3091

10777 Westheimer, Suite 1100
Houston, TX 77042
TEL (713) 260-9614
FAX (713) 260-9602

1778 Plano Road, Suite 320
Richardson, TX 75081
TEL (214) 783-0800
FAX (214) 680-2402

MARSHALL INDUSTRIES
8504 Cross Park Drive
Austin, TX 79764
TEL (512) 837-1991
FAX (512) 832-9810

Corporate Square Tech Center III
1551 North Glenville Drive
Richardson, TX 75081
TEL (214) 705-0600
FAX (214) 705-0675

10681 Haddington Drive,
Suite 160
Houston, TX 77043
TEL (713) 467-1666
FAX (713) 467-9805

MILGRAY/AUSTIN
11824 Jollyville Road,
Suite 103
Austin, TX 78759
TEL (512) 331-9961
FAX (512) 331-1070

MILGRAY/DALLAS
16610 North Dallas Parkway,
Suite 1300
Dallas, TX 75248
TEL (214) 248-1603
FAX (214) 248-0218

MILGRAY/HOUSTON
12919 S.W. Freeway, Suite 130
Stafford, TX 77477
TEL (713) 240-5360
FAX (713) 240-5404

PIONEER STANDARD ELECTRONICS
1826-D Kramer Lane
Austin, TX 78758
TEL (512) 835-4000
FAX (512) 835-9829

13765 Beta Road
Dallas, TX 75244
TEL (214) 386-7300
FAX (214) 490-6419

10530 Rockley Road
Houston, TX 77099
TEL (713) 495-4700
FAX (713) 495-5642

8200 Interstate 10 West,
Suite 705
San Antonio, TX 78230
TEL (512) 377-3440
FAX (512) 378-3626

ZEUS ELECTRONICS
3220 Commander Drive
Carrollton, TX 75006
TEL (214) 380-4330
(800) 52-HI-REL
FAX (214) 447-2222

Utah

ALL AMERICAN/UTAH
4455 South 700 East
Suite 301
Salt Lake City, UT 84107
TEL (801) 261-4210
FAX (801) 261-3885

ARROW/SCHWEBER ELECTRONICS
1946 W. Parkway Blvd.
Salt Lake City, UT 84119
TEL (801) 973-6913
FAX (801) 972-0200

INSIGHT ELECTRONICS, INC.
545 East 4500 South,
Suite E-110
Salt Lake City, UT 84107
TEL 801-288-9001
FAX 801-288-9125

MARSHALL INDUSTRIES
2355 S. 1070 West, Suite D
Salt Lake City, UT 84119
TEL (801) 973-2288
FAX (801) 973-2296

MILGRAY/UTAH
310 E. 4500 South, Suite 110
Murray, UT 84107
TEL (801) 261-2999
FAX (801) 261-0880

Washington

ALMAC/ARROW ELECTRONICS
3310 146th Place Southeast
Building B
Bellevue, WA 98007
TEL (206) 643-9992
FAX (206) 649-9709

INSIGHT ELECTRONICS, INC.
12002 115th Avenue N.E.
Kirkland, WA 98034
TEL (206) 820-8100
FAX (206) 821-2976

MARSHALL INDUSTRIES
11715 N. Creek Parkway South,
Suite 112
Bothell, WA 98011
TEL (206) 466-5747
FAX (206) 486-6964

PIONEER TECHNOLOGIES
2800 156th Avenue SE,
Suite 100
Bellevue, WA 98007
TEL (206) 644-7500
FAX (206) 644-7300

Wisconsin

ARROW/SCHWEBER ELECTRONICS
200 North Patrick Blvd.
Brookfield, WI 53045
TEL (414) 792-0150
FAX (414) 792-0156

INSIGHT ELECTRONICS, INC.
10855 West Potter Road
Suite 14
Wauwatosa, WI 53226
TEL (414) 258-5338
FAX (414) 258-5360

MARSHALL INDUSTRIES
Crossroads Corporate Center 1
20900 Swenson Drive, Suite 150
Waukesha, WI 53186
TEL (414) 797-8400
FAX (414) 797-8270

PIONEER STANDARD ELECTRONICS
120 Bishop's Way, Suite 163
Brookfield, WI 53005
TEL (414) 784-3480
FAX (414) 784-8207

Canada

ALL AMERICAN
6375 Dixie Road North,
Units 4, 5, & 6
Mississauga, Ontario, Canada
L5T 2E7
TEL (905) 670-5946
FAX (905) 670-5947

ARROW/SCHWEBER ELECTRONICS
1100 St. Regis Blvd.
Dorval, Quebec, Canada
H9P2T5
TEL (514) 421-7411
FAX (514) 421-7430

8544 Baxter Place
Burnaby, BC, Canada V5A4T8
TEL (604) 421-2333
FAX (604) 421-5030

36 Antares Drive, Unit 100
Nepean, Ontario, Canada
K2E7W5
TEL (613) 226-6903
FAX (613) 723-2018

1093 Meyerside Drive
Mississauga, Ontario, Canada
L5T1M4
TEL (905) 670-7769
FAX (905) 670-7781

INSIGHT ELECTRONICS, INC.
1405, Trans-Canada Hwy.,
Suite 200
Dorval, Quebec H9P 2V9
TEL (514) 421-7373
FAX (514) 421-0024

1 Eva Road, Suite 107
Etobicoke, Ontario M9C 4Z5
TEL (416) 622-7006
FAX (416) 622-5155

240 Catherine St., Suite 405
Ottowa, Ontario K2P 2G8
TEL (613) 233-1799
FAX (613) 233-2843

10691 Shellbridge Way,
Suite 130
Richmond, BC V6X 2W8
TEL (604) 270-3232
FAX (604) 270-3356

MARSHALL INDUSTRIES
48 Brunswick Boulevard
Pointe Claire, Quebec H9R 5P9
Canada
TEL (514) 694-8142
FAX (514) 694-6989

5285 Northam Dr.
Mississauga, Ontario L4V 1X5
Canada
TEL (905) 465-1771
FAX (905) 612-1988

MILGRAY/TORONTO
2783 Thamesgate Drive
Mississauga, Ontario L4T 1G5
Canada
TEL (905) 678-0958
FAX (905) 678-1213

MILGRAY/MONTREAL
5600 Trans Canada, Suite 209
Pointe Claire, Quebec H9R-4S2
Canada
TEL (514) 426-5900
FAX (514) 426-5836

MILGRAY/WESTERN CANADA
B2014185 Still Creek Dr.
Burnaby, B.C. V5C 6G9
TEL (604) 291-0044
FAX (604) 291-9939

PIONEER STANDARD ELECTRONICS
560 1212-31 Avenue NE
Calgary, Alberta T2E 7S8
Canada
TEL (403) 291-1988
FAX (403) 291-0740

148 York Street, Suite 209
London, Ontario N6A 1A9
Canada
TEL (519) 672-4666
FAX (519) 672-3528

3415 American Drive
Mississauga, Ontario L4V 1T6
Canada
TEL (905) 405-8300
FAX (905) 405-6425

223 Colonnade Road, Unit 12
Nepean, Ontario K2E 7K3
Canada
TEL (613) 226-8840
FAX (613) 226-6352

10711 Cambie Road, Suite 170
Richmond, B.C. V6X 3G5
Canada
TEL (604) 273-5575
FAX (604) 273-2413

Place Iberville IV
2954 Blvd. Laurier, Suite 100
Ste-Foy, Quebec G1V 4T2
Canada
TEL (418) 654-1077
FAX (418) 654-2958

520 McCaffrey Street
Ville St. Laurent, Quebec
H4T 1N1
Canada
TEL (514) 737-9700
FAX (514) 737-5212

Alabama

ELECTRONIC MARKETING ASSOC.
501 S. Memorial Parkway
Suite 106
Huntsville, AL 35802
TEL (205) 880-8050
FAX (205) 880-8054

Arizona

COMPASS MARKETING
1801 North Tatum Boulevard
Suite 101
Phoenix, AZ 85028
TEL (602) 996-0635
FAX (602) 996-0586

California

PROMERGE SALES, INC.
100 Century Center Court
Suite 710
San Jose, CA 95112
TEL (408) 467-0600
FAX (408) 467-0610

PROLINE TECHNOLOGIES
P.O. Box 1326
Healdsburg, CA 95448
TEL (707) 431-2937
FAX (707) 431-1809

HARPER & STRONG
2798 Junipero Avenue
Signal Hill, CA 90806
TEL (310) 424-3030
FAX (310) 424-6622

SILICON TECHNICAL SALES, INC.
140 Lomas Santa Fe Drive
Suite 203
Solana Beach, CA 92075
TEL (619) 793-3330
FAX (619) 793-4188

Colorado

THORSON ROCKY MOUNTAIN
7108 D South Alton Way,
Suite A
Englewood, CO 80112
TEL (303) 773-6300
FAX (303) 773-6302

Connecticut

DELTA-CONN TECHNICAL SALES
One Prestige Drive, 2nd Floor
Suite 206
Meriden, CT 06450
TEL (203) 634-8558
FAX (203) 238-1240

Florida

COMPONENT DESIGN MARKETING
800 Corporate Drive, Suite 230
Ft. Lauderdale, FL 33334
TEL (305) 492-1160
FAX (305) 492-1167

1900 S.W. 85th Avenue
North Lauderdale, FL 33068
TEL (305) 726-5444
FAX (305) 726-5155

2318 Stag Run Boulevard
Clearwater, FL 34625
TEL (813) 725-4894
FAX (813) 796-7252

7616 Southland Boulevard
Suite 103
Orlando, FL 32809
TEL (407) 240-3903
FAX (407) 240-4305

4502 West Elm Street
Tampa, FL 33614
TEL (813) 886-9721
FAX (813) 888-7816

Georgia

ELECTRONIC MARKETING ASSOC.
5855 Jimmy Carter Blvd.
Suite 190
Norcross, GA 30071
TEL (404) 448-1215
FAX (404) 446-9363

Idaho

THORSON ROCKY MOUNTAIN
1937 East Cypress Point Drive
Eagle, ID 83616
TEL (208) 939-4345
FAX (208) 939-4107

Iowa

DY-TRONIX, INC.
23 Twixt Town Road N.E.
Cedar Rapids, IA 52402
TEL (319) 377-8275
FAX (319) 377-9163

Illinois

PHASE II MARKETING
2260 Hicks Road, Suite 410
Rolling Meadows, IL 60008
TEL (847) 577-9401
FAX (847) 577-9491

Indiana

MICRO COMPONENTS, INC.
1777 East Lincoln Road
Kokomo, IN 46902
TEL (317) 455-5000
FAX (317) 455-5010

VALENTINE ASSOCIATES, INC.
1030 Summit Drive
Carmel, IN 46032
TEL (317) 846-0008
FAX (317) 846-0255

Kansas

DY-TRONIX, INC.
5001 College Boulevard,
Suite 106
Leawood, KS 66211
TEL (913) 339-6333
FAX (913) 339-9449

1999 Amidon, Suite 322
Wichita, KS 67203
TEL (316) 838-0884
FAX (316) 838-2645

Maryland

AVTEK ASSOCIATES, INC.
10632 Little Patuxent Parkway
Suite 435
Columbia, MD 21044
TEL (410) 740-5100
FAX (410) 740-5103

Massachusetts

CTC ASSOCIATES, INC.
12 Southwest Park
Westwood, MA 02090
TEL (617) 320-1818
FAX (617) 320-8282

Michigan

TRILOGY MARKETING, INC.

691 N. Squirrel Road, Suite 110
Auburn Hills, MI 48326
TEL (810) 377-4900
FAX (810) 377-4906

Minnesota

PSI

8000 Town Line Avenue S.
Suite 206
Bloomington, MN 55438
TEL (612) 944-8545
FAX (612) 944-6249

Missouri

DY-TRONIX, INC.

3407 Bridgeland Drive
Bridgeton, MO 63044
TEL (314) 291-4777
FAX (314) 291-3861

Nevada

PROLINE TECHNOLOGIES

P.O. Box 1326
Healdsburg, CA 95448
TEL (707) 431-2937
FAX (707) 431-1809

New Jersey

NORTH EAST COMPONENTS

19 Spear Road, Suite 205
Ramsey, NJ 07446
TEL (201) 825-0233
FAX (201) 934-1310

TRITEK SALES, INC.

1 Mall Drive, Suite 410
Cherry Hill, NJ 08002
TEL (609) 667-0200
FAX (609) 667-8741

New Mexico

COMPASS MARKETING

4100 Osuna Road, Suite 109
Albuquerque, NM 87109
TEL (505) 344-9990
FAX (505) 345-4848

New York

EMPIRE TECHNICAL ASSOC.

29 Fennell Street, Suite A
Skaneateles, NY 13152
TEL (315) 685-5703
FAX (315) 685-5979

349 West Commercial Street,
Suite 2920
E. Rochester, NY 14445
TEL (716) 381-8500
FAX (716) 381-0911

North Carolina

ELECTRONIC MARKETING ASSOC.

185 Windchime Ct., Suite 101
Raleigh, NC 27615
TEL (919) 847-8800
FAX (919) 848-1787

19633 Meta Road
Cornelius, NC 28031
TEL (704) 895-0043
FAX (704) 895-0730

Ohio

MILLENNIUM TECHNICAL SALES

3165 Linwood Road
Cincinnati, OH 45208
TEL (513) 871-2424
FAX (513) 871-2524

6631 Commerce Parkway,
Suite K
Dublin, OH 43017
TEL (614) 793-9545
FAX (614) 793-0256

4700 Sunray Road
Kettering, OH 45429
TEL (513) 435-8650
FAX (513) 435-8570

6519 Wilson Mills Road
Mayfield Village, OH 44143
TEL (216) 461-3500
FAX (216) 461-1335

Oklahoma

QUAD STATE SALES & MARKETING

110 W. Commercial Street,
Suite 210
Broken Arrow, OK 74013
TEL (918) 258-7723
FAX (918) 258-7653

Oregon

ELECTRONIC SOURCES, INC.

6850 SW 105th, Suite B
Beaverton, OR 97008
TEL (503) 627-0838
FAX (503) 627-0238

Pennsylvania

MILLENNIUM TECHNICAL SALES

505 Bayberry Lane
Inperial, PA 15126
TEL (412) 695-7661
FAX (412) 695-7870

South Carolina

ELECTRONIC MARKETING ASSOC.

413 Foothills Road
Greenville, SC 29609
TEL (803) 246-3884
FAX (803) 246-3929

Texas

QUAD STATE SALES & MARKETING

8310 Capital of Texas Hwy.
North Suite 365
Austin, TX 78731
TEL (512) 346-7002
FAX (512) 346-3601

12160 Abrams Road, Suite 406
Dallas, TX 75243
TEL (214) 669-8567
FAX (214) 669-8834

10565 Katy Fwy, Suite 212
Houston, TX 77024
TEL (713) 467-7749
FAX (713) 467-5942

Utah

THORSON ROCKY MOUNTAIN

5505 South 900 East,
Suite 140
Salt Lake City, UT 84117
TEL (801) 264-9665
FAX (801) 264-9881

Washington

ELECTRONIC SOURCES, INC.

1603 116th Ave, N.E., Suite 115
Bellevue, WA 98004
TEL (206) 451-3500
FAX (206) 451-1038

Wisconsin

PHASE II MARKETING

205 Bishop's Way,
Suite 220
Brookfield, WI 53005
TEL (414) 797-9986
FAX (414) 797-9935

:anada

:LARK-HURMAN
\SSOCIATES
8 Donegani, Suite 200
'ointe Claire, Quebec H9R 2V4
:anada
'EL (514) 426-0453
'AX (514) 426-0455

.08 Palladium Drive, Suite 200
:anata, Ontario K2V 1A1
:anada
'EL (613) 599-5626
'AX (613) 599-5707

6 Regan Road, Units 39, 40, 41
}rampton, Ontario L7A 1C1
:anada
'EL (905) 840-6066
'AX (905) 840-6091

'uerto Rico

:alle Hucar 38 (Bajos)
}O Sabanetas
Mercedita 00715
'uerto Rico
'EL (809) 844-3840
'AX (809) 844-3915

8

Argentina

Tucuman 1567, Officina 14
Buenos Aires
Argentina
TEL (54) 1-46-2128
FAX (54) 1-46-2128

Australia

GEC ELECTRONICS DIVISION
38 South Street
Rydalmere, N.S.W. 2116
Australia
TEL (61) 2898-7422
FAX (61) 2638-1798

Austria

CODICO
Muhlgasse 86-88
A-2380
Perchtoldsdorf/A
Austria
TEL (43) 1-863-3050
FAX (43) 1-863-0598

Belgium

ALCOM ELECTRONICS BV
Singel 3
2550 Kontich
Belgium
TEL (32) 3-4583033
FAX (32) 3-4583126

Brazil

COLGIL, INC.
Rua Marques de Itu, 306,
Conj. 94
CEP 01223-000
Sao Paulo
Brazil
TEL (55) 11-223-6954
FAX (55) 11-223-4989

Headquarters
New York
United States
TEL (212) 832-1340
FAX (212) 826-3623

HASTEC
Rua Ferreira Do Alentecj, 90/92
CEP 04728
Sao Paulo
Brazil
TEL (55) 11-522-1799
FAX (55) 11-522-5366

Bulgaria

CODICO
AP 9
5500 Lovetsch
Bulgaria 86
TEL (3592) 68-44812
FAX (3592) 68-44812

Denmark

MER-EL A/S
Ved Klaedebo 18
Post Boks 219
DK-2970 Horsholm
Denmark
TEL (45) 42-571000
FAX (45) 42-572299

Finland

ACTE NC FINLAND OY
P.O. Box 16
FIN-65611 Mustasaari
Finland
TEL (358) 0-61352690
FAX (358) 0-61352655

France

MICRO PUISSANCE
Immeuble Femto
1 Avenue de Norvege
Z.A. de Courtaboeuf B.P. 79
91943 Les Ulis Cedex
France
TEL (33) 1-69071211
FAX (33) 1-69076712

NEWTEK
8 Rue De L'Estrerel
Silic 583
94663 Rungis Cedex
France
TEL (33) 1-46872200
FAX (33) 1-46878049

Germany

INELTEK GMBH
Hauptstrasse 45
89522 Heidenhiem
Germany
TEL (49) 7321-93850
FAX (49) 7321-938595

INELTEK MITTE
Stehnweg 2
63500 Seligenstadt
Germany
TEL (49) 6182-5066
FAX (49) 6182-65953

INELTEK MURNAU
Am Fugsee 21
82418 Murnau-Riedhausen
Germany
TEL (49) 8841-47775
FAX (49) 8841-2660

INELTEK NORD
Billstrasse 30
20539 Hamburg 26
Germany
TEL (49) 78942-274
FAX (49) 78942-220

MSC VERTRIEBS GMBH
Berg-am-Laim Street 147
81673 Munchen
Germany
TEL (49) 8945-491916
FAX (49) 8945-491920

Greece

MICRELEC LTD.
339 Thivon Street
GR 12244 Aegaleo
Athens, Greece
TEL (30) 1-5395042-4
FAX (30) 1-5390269

Hong Kong

TLG ELECTRONICS, LTD.
Room 1404, Hang Shing Bldg.
363-373, Nathan Road
Yanumatei, Kowloon
Hong Kong
TEL (852) 2388-7613
FAX (852) 2783-0198

INFINITRON LTD.
B8, I/F
Shatin Ind. Centre
Siu Lek Yuzn Road
Shatin, N.T.
Hong Kong
TEL (852) 6377118
FAX (852) 6373723

Hungary

CODICO KFT
Ostrom Utca 23-25
1015 Budapest
TEL (36) 1-156-63-30
FAX (36) 1-156-43-76

India

ORIOLE SERVICES
Post Box No. 9275
5 Kurla Industrial Estate
Ghatkopar, Bombay 400 086
India
TEL (022) 5119940
FAX (022) 5115810

Ireland

ZEC SERVICES
Valleymount
Blessington
Co. Wicklow
Ireland
TEL (353) 458-64259
FAX (353) 458-64075

Israel

ISAMTEK LTD.
71 Hamelacha Street
Dan-Aviv Building
New Industrial Park
(Netanya South)
P.O. Box 8259
Netanya 42-504
Israel
TEL (972) 9-658750
FAX (972) 9-658751

Italy

LASI ELETTRONICA S.P.A.
Viale Fulvio Testi 280
20126 Milano
Italy
TEL (39) 2-66101370
FAX (39) 2-66101385

NEWTEK ITALIA SPA
Viale Cassiodoro 16
20145 Milano
Italy
TEL (39) 2-46 92 156
FAX (39) 2-46 95 197

Japan

MCM JAPAN LTD.
Santower Bldg.
2-11-22, Sangenjyaya
Setagaya-Ku, Tokyo 154
Japan
TEL (81) 3-3487-8477
FAX (81) 3-3487-8825

Tama Sales Office
Tachikawa Center Bldg.
2-22-20, Akebono Cho
Tachikawa
Tokyo 190
Japan
TEL (81) 425-22-8600
FAX (81) 425-23-8603

RYOSAN CO., LTD.
2-18-22, Soto-Kanda,
Chiyoda-ku, Tokyo
Japan
TEL (81) 3-5294-1261
FAX (81) 3-5294-1414

RYOYO ELECTRO CORP.
Konwa Bldg., 1-12-22
Tsukiji, Chuo-Ku
Tokyo 104
Japan
TEL (81) 3-3546-5011
FAX (81) 3-3546-5044

Osaka Branch
Nissin Syokuhin Bldg., 4-1-1
Nishi Nakajima, Yodogawa-ku
Osaka 532
Japan
TEL (81) 6-301-1221
FAX (81) 6-302-1002

TAKACHIHO KOHEKI CO., LTD.
1-2-8, Yotsuya
Shinjuku-Ku
Tokyo 160
Japan
TEL (81) 3-3355-6696
FAX (81) 3-3357-5034

Fukoku Seimei Bldg.
2-4, Komatsubara-Cho
Kita-Ku, Osaka 530
Japan
TEL (81) 6-313-0671
FAX (81) 6-313-3380

UNI ELECTRONICS, INC.
P.O. Box 68, Sumitomo Bldg.
2-6-1 Nishi Shinjukn
Shinjuku-ku
Tokyo 163
Japan
TEL (81) 3-3347-8878
FAX (81) 3-3347-8808

Osaka Branch
5-13-9, Mitejima
Nishi Yodogawa-ku
Osaka 555
Japan
TEL (81) 6-473-8429
FAX (81) 6-473-5513

Korea

I & C MICROSYSTEMS CO. LTD
801, 8FL, Bethel Bldg.
324-1 Yang Jae-Dong, Seocho-Ku
Seoul, Korea
TEL (82) 2-577-9131
FAX (82) 2-577-9130

SHINHWA CORPORATION
2F, The Christian Literature,
Society of Korea Bldg.
169-1, Samsung-Dong,
Kangnam-Ku
Seoul, Korea
TEL (82) 2-554-6431
FAX (82) 2-554-7649

SEGYUNG TECHCELL CORP.
Dansan Nonhyun
Bldg. 4F 270-45
Nonhyun-Dong, Kangnan-Ku
Seoul, Korea
TEL (82) 2-515-7477
FAX (82) 2-515-8889

UNIQUEST KOREA
Suite 1110 Daejong Bldg.
143-48, Samsung-Dong,
Kangnam-Gu
Seoul, Korea
TEL (82) 2-562-8805
FAX (82) 2-562-6646

Mexico

ADELSA
Calzada Del Rio #7915-2
Residencial Antares, C.P. 32420
Ciudad Juarez, Chihuahua
Mexico
TEL (011) 521-625-5234
FAX (011) 521-625-5234

Club Cuicacalli #66
CTO Cronistas
Zona Azul CD Satelite
Naucalpan De Juarez
C.P. 53100 EDO de Mexico
Mexico
TEL (525) 374-0981
FAX (525) 374-0997

Netherlands

ALCOM ELECTRONICS BV
Essebaan 1
2908 LG Capelle Aan Den Yssel
Netherlands
TEL (31) 10-4519533
FAX (31) 10-4586482

New Zealand

APEX ELECTRONICS LTD.
175 Vivian Street
Wellington, New Zealand
TEL (64) 4-3853404
FAX (64) 4-3853483

GEC ELECTRONICS
5 Reliable Way
Penrose Auckland
New Zealand
TEL (64) 9-526-0107
FAX (64) 9-525-7923

Norway

NC NORDCOMP AS
Vestvollveien 10C
2020 Skedsmokorset
Norway
TEL (47) 63-879330
FAX (47) 63-879000

Poland

CODICO
Jl Bora Komorowskiego 6/17
36-300 Grudziadz
Poland
TEL (48) 51-23223
FAX (48) 51-23223

Portugal

ANATRONIC-PORTUGAL S.A.
Rua Padre Antonio Vieira,
No 31B
Povoa De Santo Adriao - 2675
Odivelas
Portugal
TEL (351) 1-9376267
FAX (351) 1-9371834

Romania

CODICO
BD Decebal NR 11 BLS 14
SC 2, Apt. 41, Sect. 3
RO-Bucharest
TEL (40) 1-3210431
FAX (40) 1-3210431

Singapore

NUCLEUS ELECTRONICS PTE LTD.
5001 Beach Road #18-03
Golden Mile Tower
Singapore 0719
TEL (65) 297-6883
FAX (65) 297-6882

SINGASOF COMPUTERS PTE., LTD.
2 Kallang Pudding Road #09-04
Mactech Industrial Bldg.
Singapore 1334
TEL (65) 747-3012
FAX (65) 747-5279

Slovakia

CODICO
Nad Santoskoce 11
CZ-120 00 Prag
Slovakia
TEL (42) 2 545 825
FAX (42) 2 537 950

Slovenia

CODICO
Muhlgasse 86-88
A-2380 Perchtoklsclorf
Slovenia
TEL (43) 1 86305 54
FAX (43) 1 86305 98

South Africa

ADVANCED SEMICONDUCTOR DEVICES (PTY), LTD.
P.O. Box 3853
Rivonia 2128
South Africa
TEL (27) 11-444-2333
FAX (27) 11-444-1706

Spain

ANATRONIC S.A.
AVDA. De Valladolid, 27
28008 Madrid
Spain
TEL (34) 1-542-44-55
FAX (34) 1-559-69-75
Bailen, 176. Entresuelo 1o
08037 Barcelona
Spain
TEL (34) 3-458-19-06
FAX (34) 3-458-71-28

Las Mercedes 25 3o Dpto. 1
48930 Las Arenas-Vizcaya
Spain
TEL (34) 4-463-60-66
FAX (34) 4-463-42-35

Sweden

KELTECH COMPONENTS AB
Box 505 Kemistvagen 10 A
S-183 25
Taby
Sweden
TEL (46) 86308590
FAX (46) 87562143

Switzerland

ANATEC AG
Sumpfstrasse 7
CH 6300
ZUG
Switzerland
TEL (41) 42-412441
FAX (41) 42-413124

Taiwan

ALLREACH ENTERPRISE CO., LTD.
6F, No. 50-1, Sec. 1
Hsin Sheng South Road
Taipei, Taiwan
R.O.C.
TEL (886) 2-321-4229
FAX (886) 2-321-5849

APPLIED COMPONENT TECHNOLOGY CORP.
8F, No. 233-1 Pao Chiao Road
Hsin Tien City
Taipei Hsien
Taiwan, R.O.C.
TEL (886) 2-917-0858
FAX (886) 2-917-1895

NEOLEC INTERNATIONAL INC.
3F, Jen Jen Bldg
No. 29, Jen Ai Road, Sec. 3,
Taipei 106
Taiwan, R.O.C.
TEL (886) 2-778-8716
FAX (886) 2-778-6076

PRINCETON TECHNOLOGY CORP.

2F, No. 233-1, Bao Chiao Road
Hsin Tien, Taipei Hsien
Taiwan, R.O.C.
TEL (886) 2-917-8856
FAX (886) 2-917-3836

TOPTREND TECHNOLOGIES CORP.

8F, No. 669, Sec. 5
Chung Hsiao East Road
Taipei, Taiwan
R.O.C.
TEL (886) 2-769-6220
FAX (886) 2-769-6228

Thailand

SEMIKON COMPANY, LTD.

4F, Room 406, Phansak Bldg.
Phetburi Road
138/1 Rajthevee, Bangkok,
10400
Thailand
TEL (662) 215-0760
FAX (662) 215-6857

Turkey

AZTECH ELECTRONIK, LTD.

Kaptanpasa Sok No. 25/2
06700 G.O.P.
Ankara, Turkey
TEL (90) 312-447-0384
FAX (90) 312-447-0387

United Kingdom

GD TECHNIK, LTD.

Tudor House
24 High Street
Twyford, Berks RG10 9AG
England
TEL (44)1734-342277
FAX (44) 1734-342896

INSIGHT TECHNOLOGY

Thame Park Road
Thame, Oxfordshire, OX9 3XD
TEL (44) 1844-261686
FAX (44) 1844-261601