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

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

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

Contact us

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

Features

- Utilizes the AVR® RISC Architecture
- AVR – High-performance and Low-power RISC Architecture
  - 121 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers + Peripheral Control Registers
  - Up to 6 MIPS Throughput at 6 MHz
- Data and Nonvolatile Program Memory
  - 128K Bytes of In-System Programmable Flash
    Endurance: 1,000 Write/Erase Cycles
  - 4K Bytes Internal SRAM
  - 4K Bytes of In-System Programmable EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program and EEPROM Data Security
  - SPI Interface for In-System Programming
- Peripheral Features
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
  - Programmable Serial UART
  - Master/Slave SPI Serial Interface
  - Real-time Counter (RTC) with Separate Oscillator
  - Two 8-bit Timer/Counters with Separate Prescaler and PWM
  - Expanded 16-bit Timer/Counter System with Separate Prescaler, Compare, Capture Modes and Dual 8-, 9-, or 10-bit PWM
  - Programmable Watchdog Timer with On-chip Oscillator
  - 8-channel, 10-bit ADC
- Special Microcontroller Features
  - Low-power Idle, Power-save and Power-down Modes
  - Software Selectable Clock Frequency
  - External and Internal Interrupt Sources
- Specifications
  - Low-power, High-speed CMOS Process Technology
  - Fully Static Operation
- Power Consumption at 4 MHz, 3V, 25°C
  - Active: 5.5 mA
  - Idle Mode: 1.6 mA
  - Power-down Mode: < 1 µA
- I/O and Packages
  - 32 Programmable I/O Lines, 8 Output Lines, 8 Input Lines
  - 64-lead TQFP
- Operating Voltages
  - 2.7 - 3.6V for ATmega103L
  - 4.0 - 5.5V for ATmega103
- Speed Grades
  - 0 - 4 MHz for ATmega103L
  - 0 - 6 MHz for ATmega103

8-bit AVR® Microcontroller with 128K Bytes In-System Programmable Flash

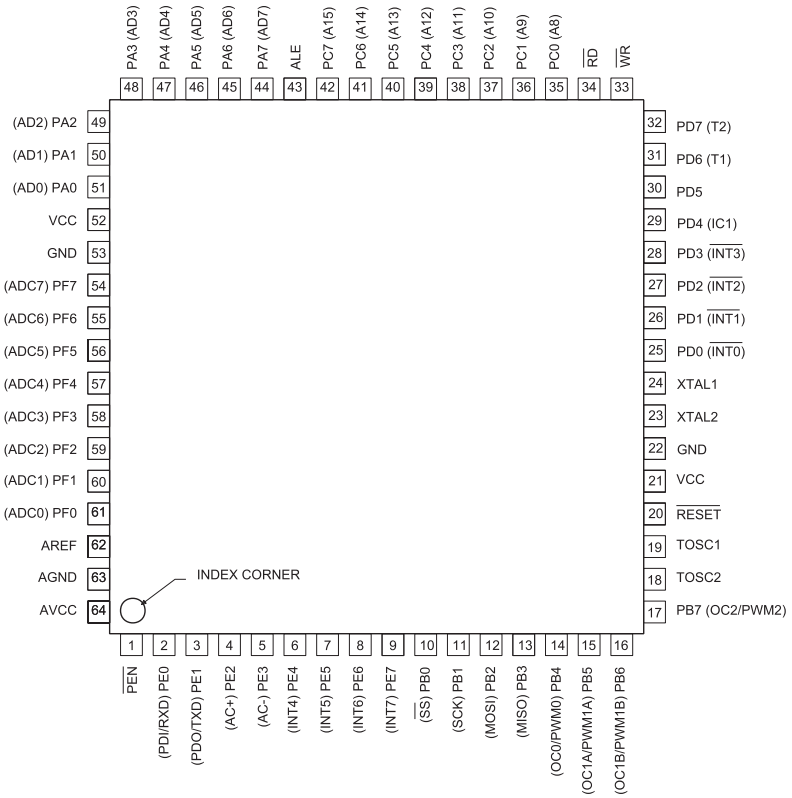
ATmega103
ATmega103L

Note: Not recommended in new designs.

Rev. 0945I–AVR–02/07

Pin Configuration

TQFP



Description

The ATmega103(L) is a low-power, CMOS, 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega103(L) 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 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.

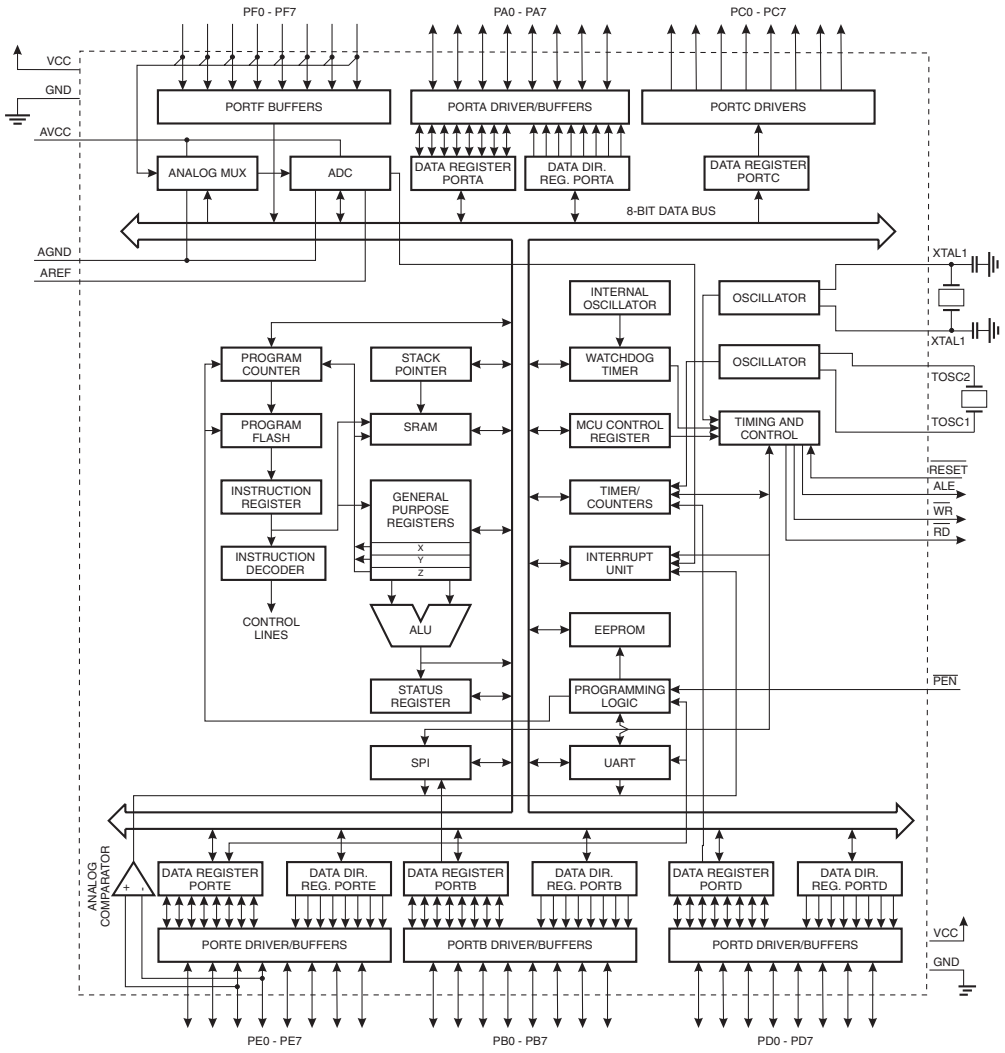
The ATmega103(L) provides the following features: 128K bytes of In-System Programmable Flash, 4K bytes EEPROM, 4K bytes SRAM, 32 general purpose I/O lines, 8 input lines, 8 output lines, 32 general purpose working registers, Real Time Counter (RTC), 4 flexible Timer/Counters with compare modes and PWM, UART, programmable Watchdog Timer with internal Oscillator, an SPI serial port and 3 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. In Power-save mode, the Timer Oscillator continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the Program memory to be reprogrammed In-System through a serial interface or by a conventional nonvolatile memory programmer. By combining an 8-bit RISC CPU with a large array of ISP Flash on a monolithic chip, the Atmel ATmega103(L) is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The ATmega103(L) 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.

Block Diagram

Figure 1. The ATmega103(L) Block Diagram



Pin Descriptions

VCC Supply voltage.

GND Ground.

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

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

The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

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

Port B also serves the functions of various special features.

The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port C (PC7..PC0) Port C is an 8-bit output port. The Port C output buffers can sink 20 mA.

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

Since Port C is an output only port, the Port C pins are *not* tri-stated when a reset condition becomes active.

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

Port D also serves the functions of various special features.

The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E (PE7..PE0) Port E is an 8-bit bi-directional I/O port with internal pull-up resistors. The Port E output buffers can sink 20 mA. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated.

Port E also serves the functions of various special features.

The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running

Port F (PF7..PF0) Port F is an 8-bit input port. Port F also serves as the analog inputs for the ADC.

$\overline{\text{RESET}}$ Reset input. An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 50 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

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

XTAL2 Output from the inverting Oscillator amplifier.

TOSC1 Input to the inverting Timer/Counter Oscillator amplifier.

TOSC2 Output from the inverting Timer/Counter Oscillator amplifier.

$\overline{WR}$ External SRAM write strobe

$\overline{RD}$ External SRAM read strobe

ALE 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.

AVCC Supply voltage for Port F, including ADC. The pin must be connected to VCC when not used for the ADC. See “ADC Noise Canceling Techniques” on page 82 for details when using the ADC.

AREF AREF is the analog reference input for the ADC converter. For ADC operations, a voltage in the range AGND to AVCC must be applied to this pin.

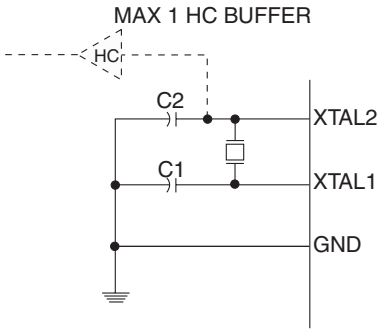
AGND If the board has a separate analog ground plane, this pin should be connected to this ground plane. Otherwise, connect to GND.

$\overline{PEN}$ PEN is a programming enable pin for the Serial Programming mode. By holding this pin low during a Power-on Reset, the device will enter the Serial Programming mode. $\overline{PEN}$ has no function during normal operation.

Clock Options

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.

Figure 2. Oscillator Connections

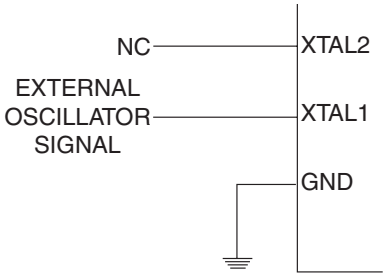


Note: When using the MCU Oscillator as a clock for an external device, an HC buffer should be connected as indicated in the figure.

External Clock

To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 3.

Figure 3. External Clock Drive Configuration

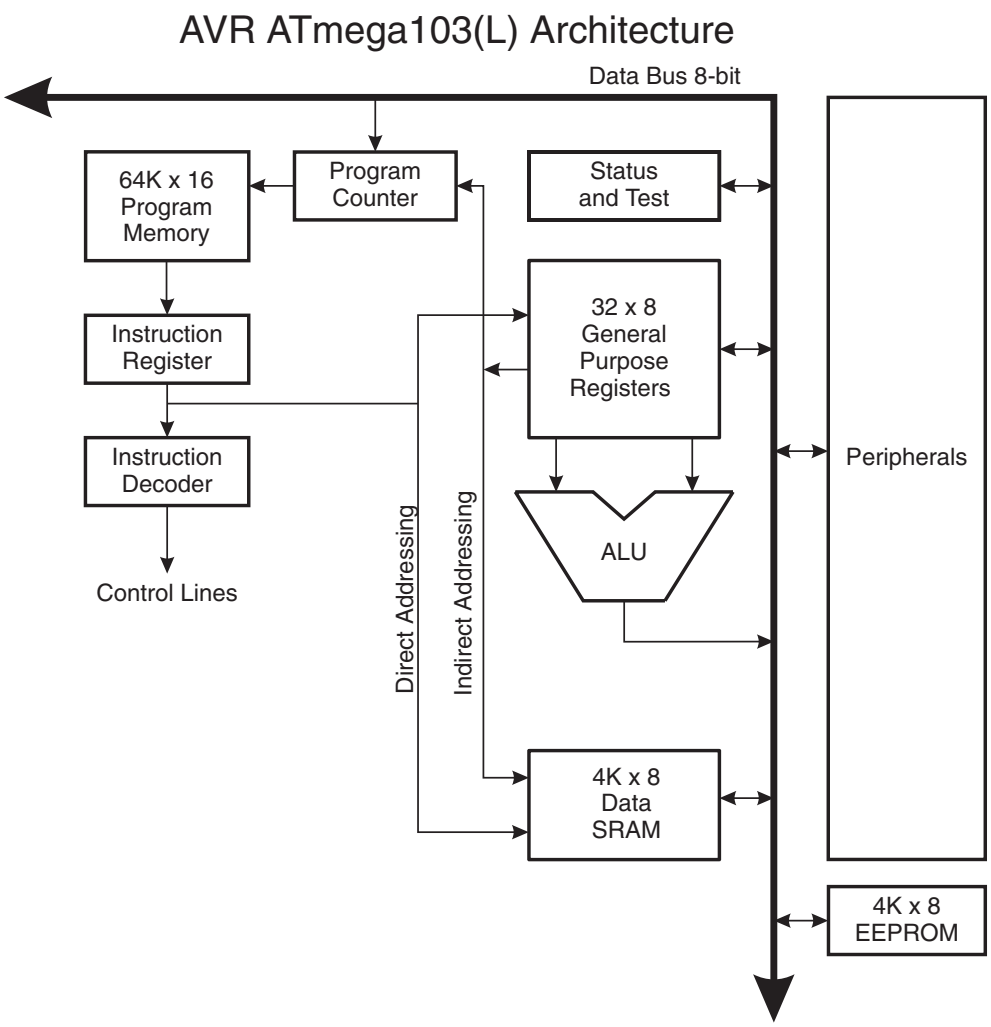


Timer Oscillator

For the Timer Oscillator pins, TOSC1 and TOSC2, the crystal is connected directly between the pins. No external capacitors are needed. The Oscillator is optimized for use with a 32,768 Hz watch crystal. Applying an external clock source to TOSC1 is not recommended.

Architectural Overview

Figure 4. The ATmega103(L) AVR RISC Architecture



The AVR uses a Harvard architecture concept – with separate memories and buses for program and data. The Program memory is accessed with a single-level pipeline. 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 Programmable Flash memory. With a few exceptions, 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 4000 bytes data SRAM can be easily accessed through the five different addressing modes supported in the AVR 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 sep-

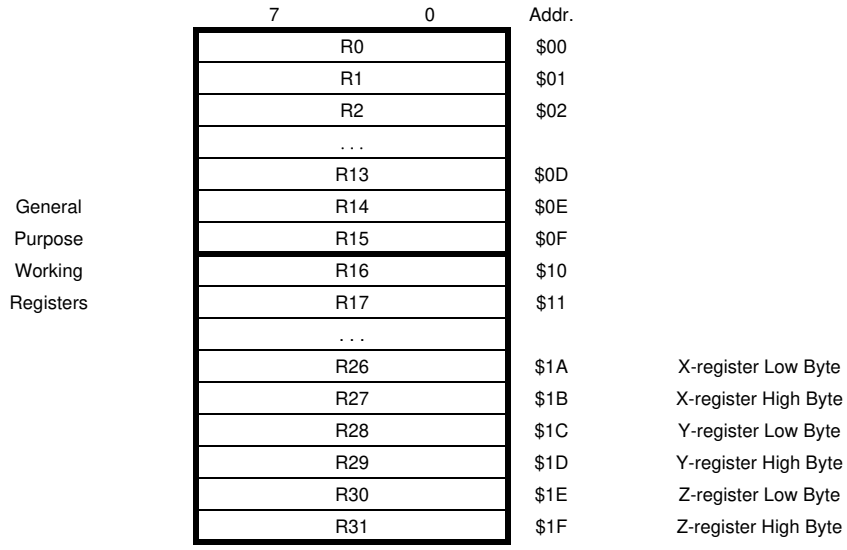
arate Interrupt Vector in the Interrupt Vector table at the beginning of the Program memory. The different interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The memory spaces in the AVR architecture are all linear and regular memory maps.

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 are 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.

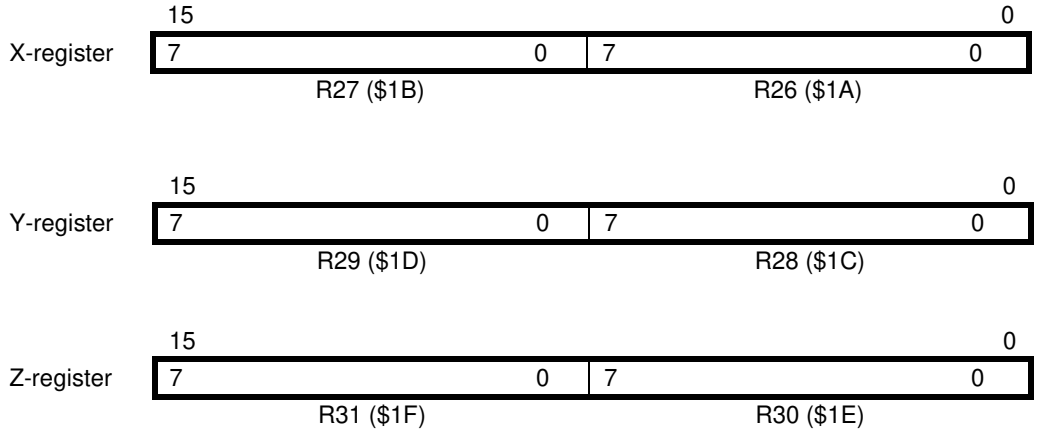
As shown in Figure 5, each register is also assigned a Data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y-, and Z-registers can be set to index any register in the file.

The 4K bytes of SRAM available for general data are implemented as addresses $0060 to $0FFF.

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. 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).

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, logical and bit functions.

ISP Flash Program Memory

The ATmega103(L) contains 128K bytes of On-chip In-System Programmable Flash memory for program storage. Since all instructions are single or double 16-bit words, the Flash is organized as 64K x 16. The Flash memory has an endurance of at least 1000 write/erase cycles.

Constant tables can be allocated in the entire Program memory space (see the LPM – Load Program Memory and ELPM – Extended Load Program Memory instruction descriptions).

SRAM Data Memory

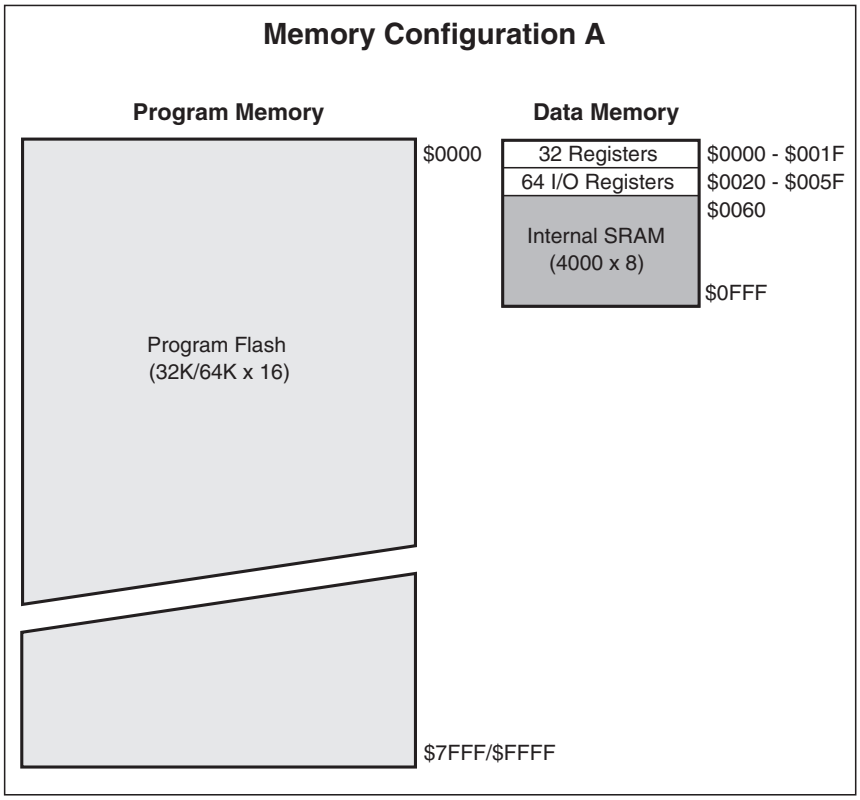
The ATmega103(L) supports two different configurations for the SRAM Data memory as listed in Table 1.

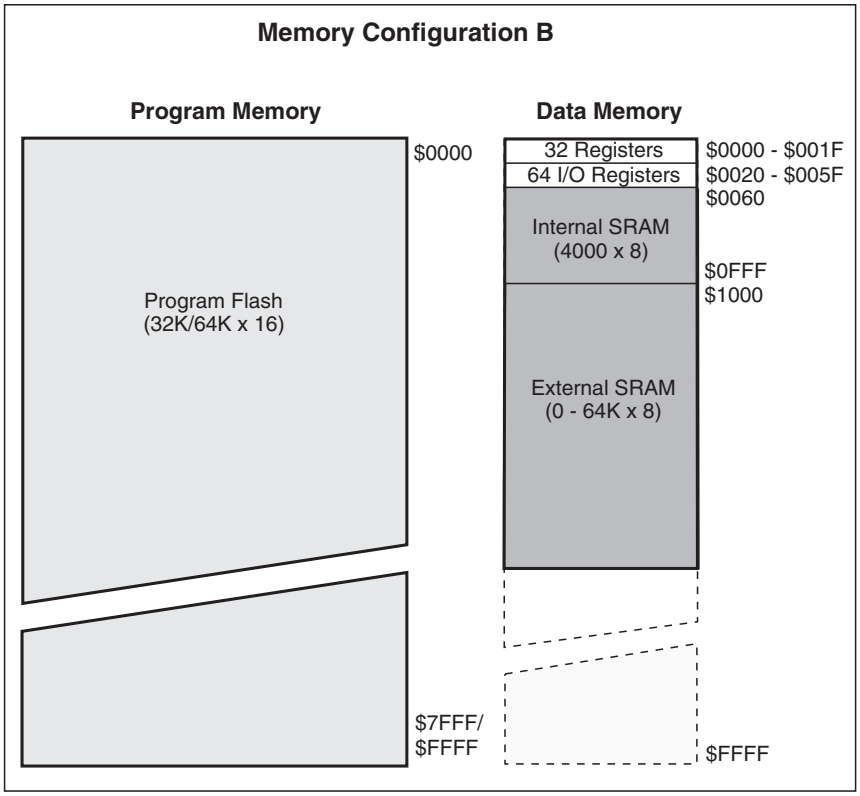
Table 1. Memory Configurations

Configuration	Internal SRAM Data Memory	External SRAM Data Memory
A	4000	None
B	4000	up to 64K

Note: When using 64K of external SRAM, 60K will be available.

Figure 7. Memory Configurations





The 4096 first Data memory locations address both the Register File, the I/O memory and the internal Data SRAM. The first 96 locations address the Register File and I/O memory, and the next 4000 locations address the internal Data SRAM.

An optional external Data SRAM can be used with the ATmega103(L). This SRAM will occupy an area in the remaining address locations in the 64K address space. This area starts at the address following the internal SRAM. If a 64K external SRAM is used, 4K of the external memory is lost as the addresses are occupied by internal memory.

When the addresses accessing the SRAM memory space exceeds the internal Data memory locations, the external Data SRAM is accessed using the same instructions as for the internal Data memory access. When the internal Data memories are accessed, the read and write strobe pins ($\overline{RD}$ and $\overline{WR}$) are inactive during the whole access cycle. External SRAM operation is enabled by setting the SRE bit in the MCUCR Register.

Accessing external SRAM takes one additional clock cycle per byte compared to access of the internal SRAM. This means that the commands LD, ST, LDS, STS, PUSH and POP take one additional clock cycle. If the Stack is placed in external SRAM, interrupts, subroutine calls and returns take two clock cycles extra because the 2-byte Program Counter is pushed and popped. When external SRAM interface is used with wait state, two additional clock cycles are used per byte. This has the following effect: Data transfer instructions take two extra clock cycles, whereas interrupt, subroutine calls and returns will need four clock cycles more than specified in the “Instruction Set Summary” on page 135.

The five different addressing modes for the Data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

The Indirect with Displacement mode features 63 address locations reached 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.

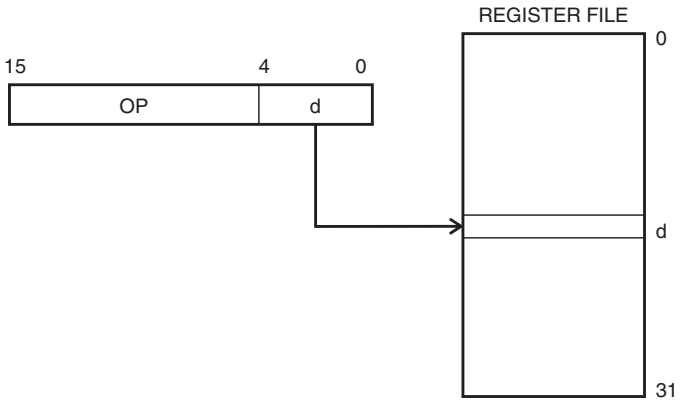
The entire Data address space including the 32 general purpose working registers and the 64 I/O Registers are all accessible through all these addressing modes. See the next section for a detailed description of the different addressing modes.

Program and Data Addressing Modes

The ATmega103(L) AVR RISC microcontroller supports powerful and efficient addressing modes for access to the Program memory (Flash) and Data memory (SRAM, Register File and I/O memory). 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

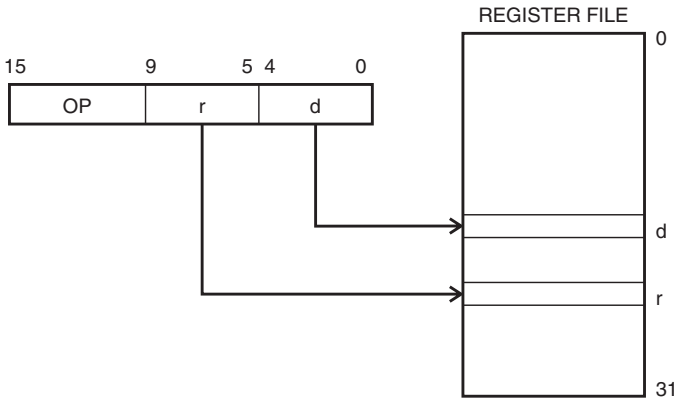
Figure 8. Direct Single Register Addressing



The operand is contained in register d (Rd).

Register Direct, Two Registers Rd and Rr

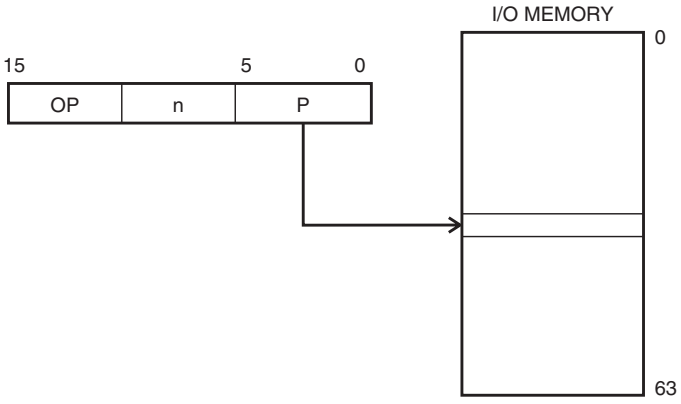
Figure 9. Direct Register Addressing, Two Registers



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

I/O Direct

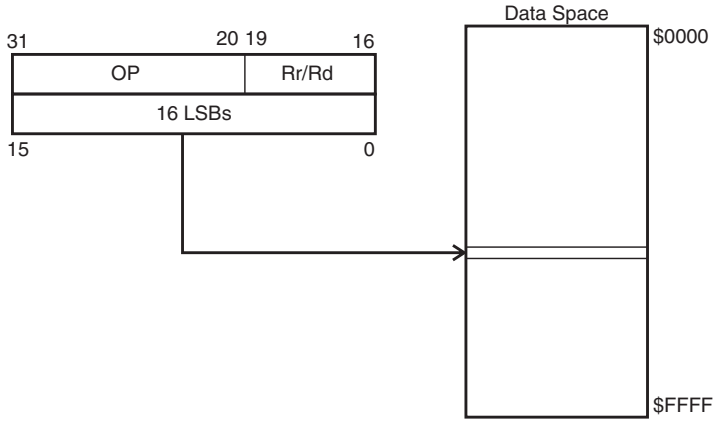
Figure 10. I/O Direct Addressing



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

Data Direct

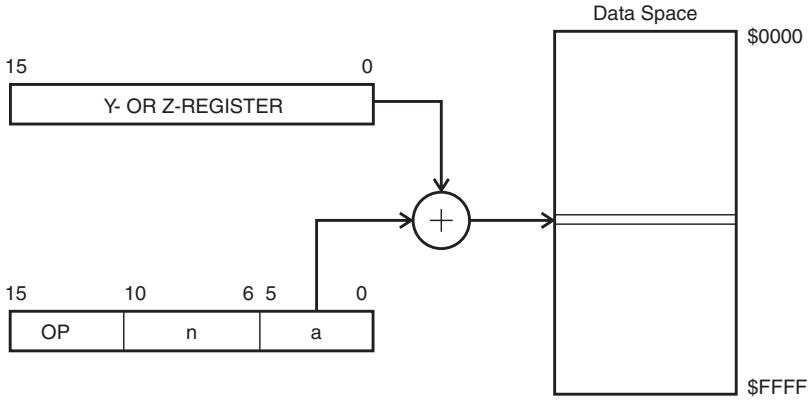
Figure 11. Direct Data Addressing



A 16-bit Data address is contained in the 16 LSBs of a 2-word instruction. Rd/Rr specify the destination or source register.

Data Indirect with Displacement

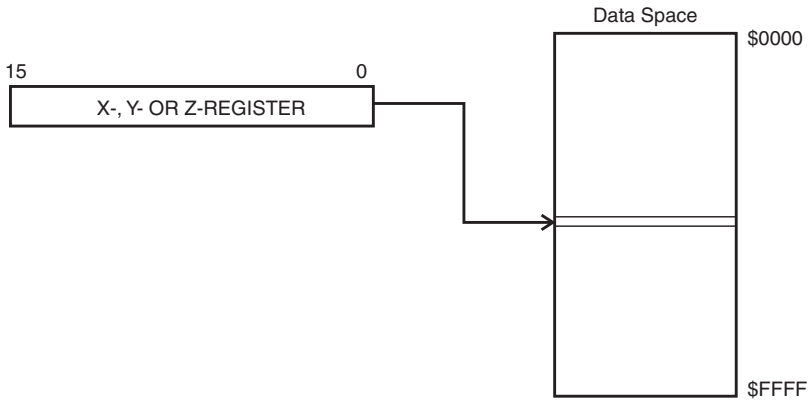
Figure 12. Data Indirect with Displacement



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

Data Indirect

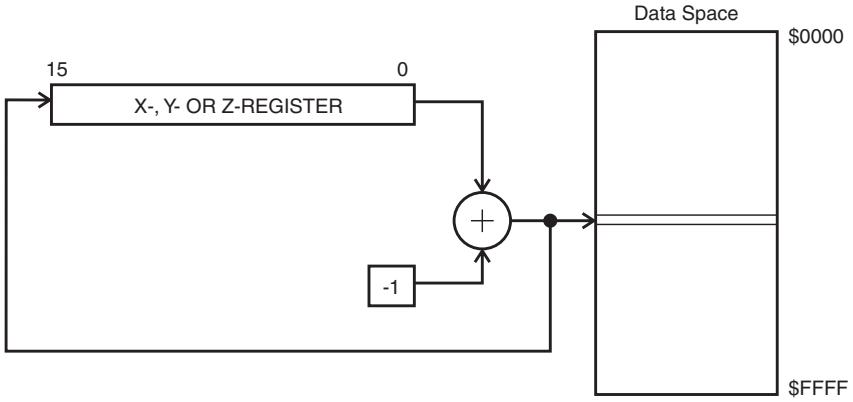
Figure 13. Data Indirect Addressing



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

Data Indirect with Pre-decrement

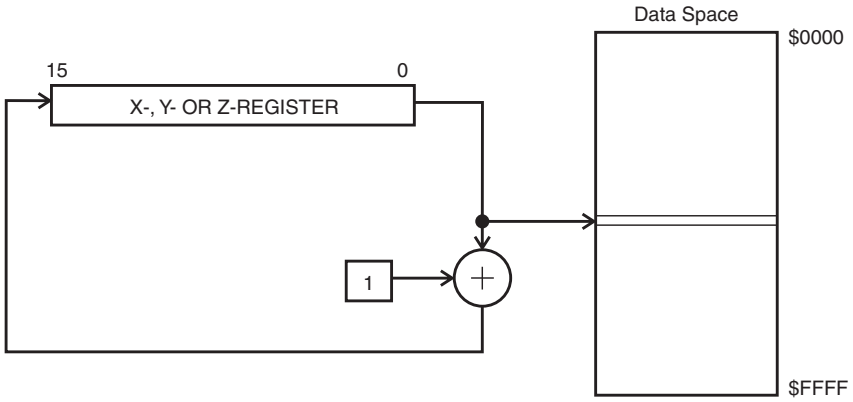
Figure 14. Data Indirect Addressing with Pre-decrement



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.

Data Indirect with Post-increment

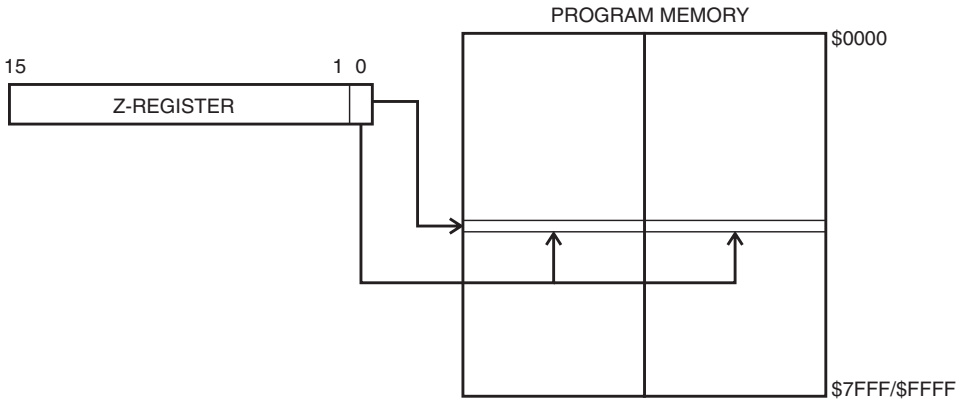
Figure 15. Data Indirect Addressing with Post-increment



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

Constant Addressing Using the LPM and ELPM Instructions

Figure 16. Code Memory Constant Addressing

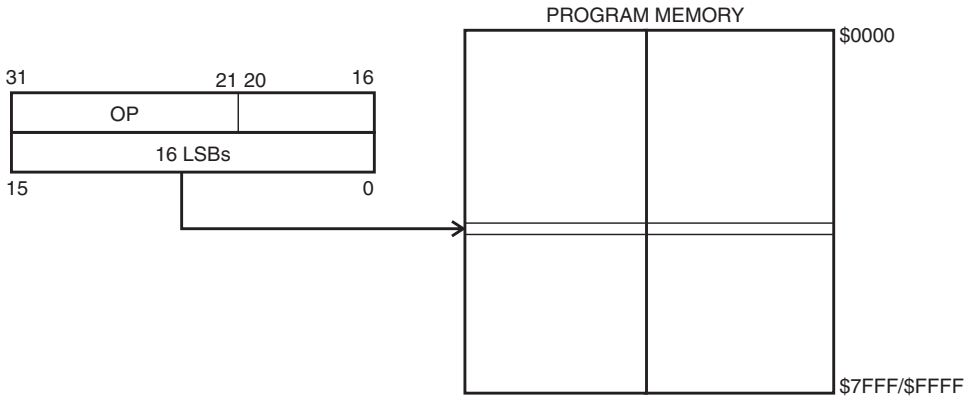


Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 32K), LSB selects Low Byte if cleared (LSB = 0) or High Byte if set (LSB =

1). If ELPM is used, LSB of the RAM Page Z register (RAMPZ) is used to select low or high memory page (RAMPZ0 = 0: Low Page, RAMPZ0 = 1: High Page).

Direct Program Address, JMP and CALL

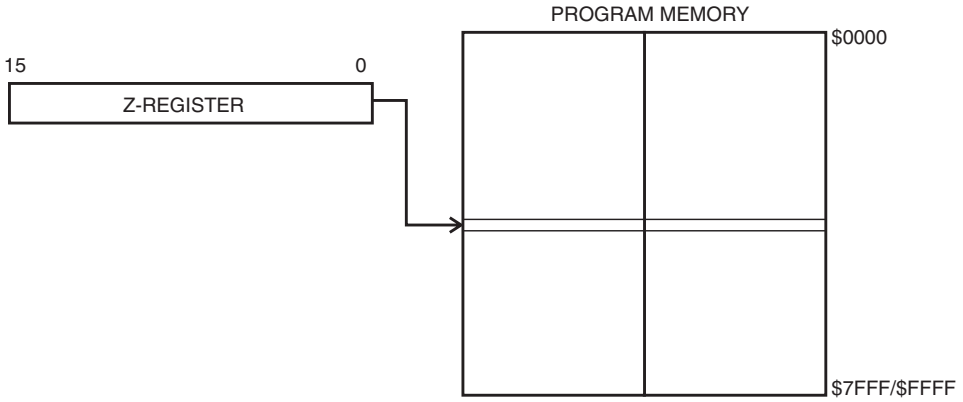
Figure 17. Direct Program Memory Addressing



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

Indirect Program Addressing, IJMP and ICALL

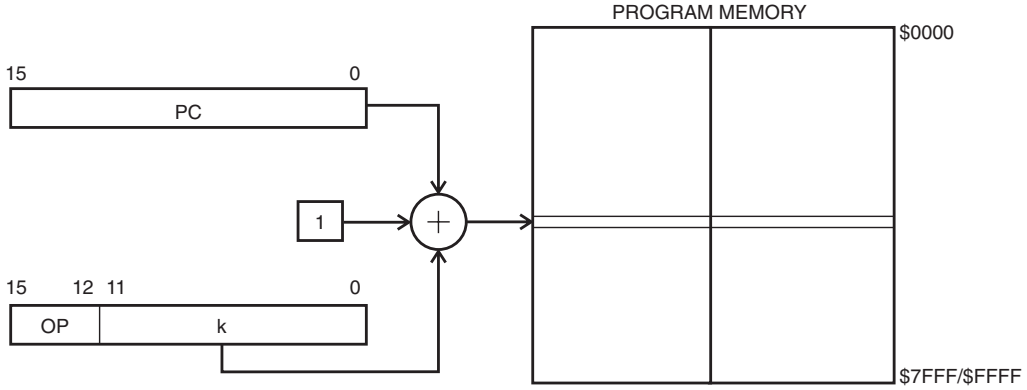
Figure 18. Indirect Program Memory Addressing



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 19. Relative Program Memory Addressing



Program execution continues at address PC + k + 1. The relative address k is -2048 to 2047.

EEPROM Data Memory

The EEPROM memory 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 57 specifying the EEPROM Address Register, the EEPROM Data Register and the EEPROM Control Register.

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 20 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 20. The Parallel Instruction Fetches and Instruction Executions

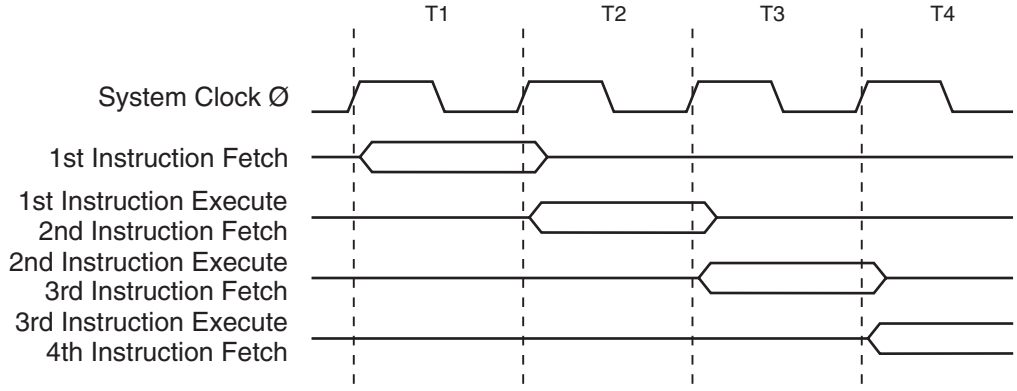
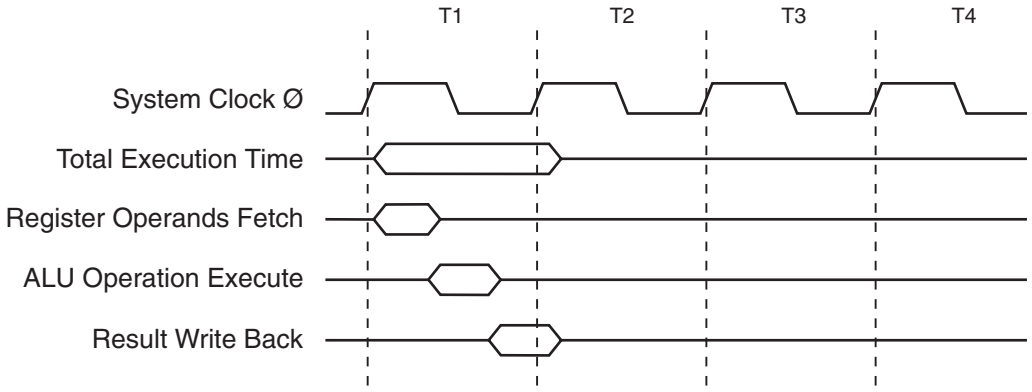


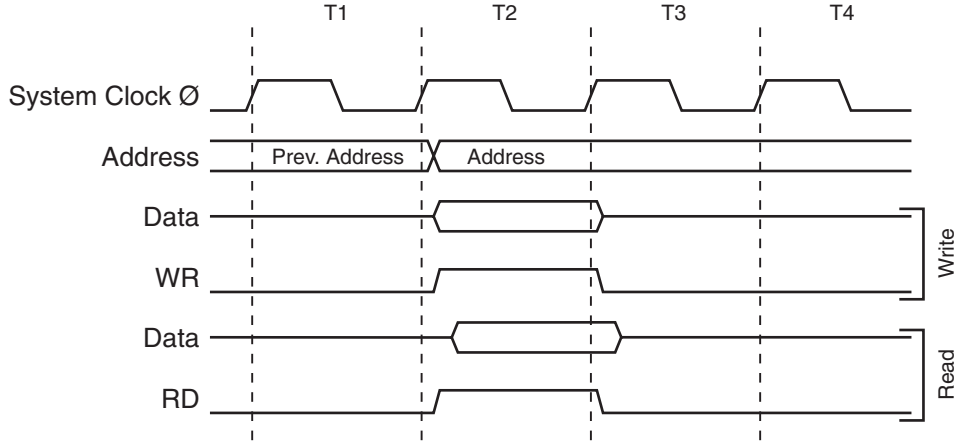
Figure 21 shows the internal timing concept for the Register File. In a single clock cycle, an ALU operation using two register operands is executed and the result is stored back to the destination register.

Figure 21. Single Cycle ALU Operation



The internal Data SRAM access is performed in two System Clock cycles as described in Figure 22.

Figure 22. On-chip Data SRAM Access Cycles



See “Interface to External SRAM” on page 84. for a description of the access to the external SRAM.

I/O Memory

The I/O space definition of the ATmega103(L) is shown in Table 2.

Table 2. ATmega103(L) I/O Space

I/O Address (SRAM Address)	Name	Function
$3F ($5F)	SREG	Status REGister
$3E ($5E)	SPH	Stack Pointer High
$3D ($5D)	SPL	Stack Pointer Low
$3C ($5C)	XDIV	XTAL Divide Control Register
$3B ($5B)	RAMPZ	RAM Page Z Select Register
$3A ($5A)	EICR	External Interrupt Control Register
$39 ($59)	EIMSK	External Interrupt MaSK Register
$38 ($58)	EIFR	External Interrupt Flag Register
$37 ($57)	TIMSK	Timer/Counter Interrupt MaSK Register
$36 ($56)	TIFR	Timer/Counter Interrupt Flag Register
$35 ($55)	MCUCR	MCU General Control Register
$34 ($54)	MCUSR	MCU Status Register
$33 ($53)	TCCR0	Timer/Counter0 Control Register
$32 ($52)	TCNT0	Timer/Counter0 (8-bit)
$31 ($51)	OCR0	Timer/Counter0 Output Compare Register
$30 ($50)	ASSR	Asynchronous Mode Status Register
$2F ($4F)	TCCR1A	Timer/Counter1 Control Register A
$2E ($4E)	TCCR1B	Timer/Counter1 Control Register B

Table 2. ATmega103(L) I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$2D ($4D)	TCNT1H	Timer/Counter1 High Byte
$2C ($4C)	TCNT1L	Timer/Counter1 Low Byte
$2B ($4B)	OCR1AH	Timer/Counter1 Output Compare Register A High Byte
$2A ($4A)	OCR1AL	Timer/Counter1 Output Compare Register A Low Byte
$29 ($49)	OCR1BH	Timer/Counter1 Output Compare Register B High Byte
$28 ($48)	OCR1BL	Timer/Counter1 Output Compare Register B Low Byte
$27 ($47)	ICR1H	Timer/Counter1 Input Capture Register High Byte
$26 ($46)	ICR1L	Timer/Counter1 Input Capture Register Low Byte
$25 ($45)	TCCR2	Timer/Counter2 Control Register
$24 ($44)	TCNT2	Timer/Counter2 (8-bit)
$23 ($43)	OCR2	Timer/Counter2 Output Compare Register
$21 ($41)	WDTCR	Watchdog Timer Control Register
$1F ($3F)	EEARH	EEPROM Address Register High
$1E ($3E)	EEARL	EERPOM Address Register Low
$1D ($3D)	EEDR	EEPROM Data Register
$1C ($3C)	EECR	EEPROM Control Register
$1B ($3B)	PORTA	Data Register, Port A
$1A ($3A)	DDRA	Data Direction Register, Port A
$19 ($39)	PINA	Input Pins, Port A
$18 ($38)	PORTB	Data Register, Port B
$17 ($37)	DDRB	Data Direction Register, Port B
$16 ($36)	PINB	Input Pins, Port B
$15 ($35)	PORTC	Data Register, Port C
$12 ($32)	PORTD	Data Register, Port D
$11 ($31)	DDRD	Data Direction Register, Port D
$10 ($30)	PIND	Input Pins, Port D
$0F ($2F)	SPDR	SPI I/O Data Register
$0E ($2E)	SPSR	SPI Status Register
$0D ($2D)	SPCR	SPI Control Register
$0C ($2C)	UDR	UART I/O Data Register
$0B ($2B)	USR	UART Status Register
$0A ($2A)	UCR	UART Control Register
$09 ($29)	UBRR	UART Baud Rate Register
$08 ($28)	ACSR	Analog Comparator Control and Status Register
$07 ($27)	ADMUX	ADC Multiplexer Select Register

Table 2. ATmega103(L) I/O Space (Continued)

I/O Address (SRAM Address)	Name	Function
$06 ($26)	ADCSR	ADC Control and Status Register
$05 ($25)	ADCH	ADC Data Register High
$04 ($24)	ADCL	ADC Data Register Low
$03 ($23)	PORTE	Data Register, Port E
$02 ($22)	DDRE	Data Direction Register, Port E
$01 ($21)	PINE	Input Pins, Port E
$00 ($20)	PINF	Input Pins, Port F

Note: Reserved and unused locations are not shown in the table.

All the different ATmega103(L) I/Os and peripherals are placed in the I/O space. The different I/O locations are directly accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. I/O Registers within the address range $00 - $1F are directly bit-accessible using the SBI and CBI instructions. In these registers, the value of single bits can be checked by using the SBIS and SBIC instructions. Refer to the “Instruction Set Summary” on page 135 for more details. When using the I/O specific instructions IN and OUT, the I/O Register address $00 - $3F are used. When addressing I/O Registers as SRAM, $20 must be added to this address. All I/O Register addresses throughout this document are shown with the SRAM address in parentheses.

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

Some of the Status Flags are cleared by writing a logical “1” to them. Note that the CBI and SBI instructions will operate on all bits in the I/O Register, writing a one back into any flag read as set, thus clearing the flag. The CBI and SBI instructions work with registers $00 to $1F only.

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

Status Register – SREG

The AVR Status Register (SREG) at I/O space location $3F ($5F) is defined as:

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	I	T	H	S	V	N	Z	C	SREG
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
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 separate control registers. If the Global Interrupt Enable Register is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. 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 cop-

ied 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 on page 135 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 on page 135 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 on page 135 for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result from an arithmetical or logical operation. See the Instruction set description on page 135 for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result from an arithmetical or logical operation. See the instruction set description on page 135 for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetical or logical operation. See the instruction set description on page 135 for detailed information.

Note that the Status Register is not automatically stored when entering an interrupt routine or restored when returning from an interrupt routine. This must be handled by software.

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 ($5E) and $3D ($5D). As the ATmega103(L) supports up to 64K bytes memory, all 16 bits are used.

Bit	15	14	13	12	11	10	9	8	
$3E ($5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
$3D ($5D)	**SP7**	**SP6**	**SP5**	**SP4**	**SP3**	**SP2**	**SP1**	**SP0**	**SPL**
	7	6	5	4	3	2	1	0	
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 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 must be set to point above $60. The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction and it is decremented by 2 when an address is pushed onto the Stack with subroutine calls and interrupts. The Stack Pointer is incremented by 1 when data is popped from the Stack with the POP

instruction and it is incremented by 2 when an address is popped from the Stack with return from subroutine RET or return from interrupt RETI.

RAM Page Z Select Register – RAMPZ

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	–	–	–	–	–	–	–	**RAMPZ0**	**RAMPZ**
Read/Write	R	R	R	R	R	R	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The RAMPZ Register is normally used to select which 64K RAM page is accessed by the Z pointer. As the ATmega103(L) does not support more than 64K of SRAM memory, this register is used only to select which page in the Program memory is accessed when the ELPM instruction is used. The different settings of the RAMPZ0 bit have the following effects:

RAMPZ0 = 0: Program memory address $0000 - $7FFF (lower 64K bytes) is accessed by ELPM

RAMPZ0 = 1: Program memory address $8000 - $FFFF (higher 64K bytes) is accessed by ELPM

Note that LPM is not affected by the RAMPZ setting.

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 ($55)	**SRE**	**SRW**	**SE**	**SM1**	**SM0**	–	–	–	**MCUCR**
Read/Write	R/W	R/W	R/W	R/W	R/W	R	R	R	
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), and A8 - 15 (Port C) are activated as the alternate pin functions. Then the SRE bit overrides any pin direction settings in the respective Data Direction Registers. 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 a three-cycle scheme. See Figure 51 on page 85 and Figure 52 on page 85.

- **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 programmer's purpose, it is recommended to set the Sleep Enable (SE) bit just before the execution of the SLEEP instruction.

- **Bits 4, 3 – SM1/SM0: Sleep Mode Select Bits 1 and 0**

This bit selects between the three available sleep modes as shown in Table 3.

Table 3. Sleep Mode Select

SM1	SM0	Sleep Mode
0	0	Idle mode
0	1	Reserved
1	0	Power-down
1	1	Power-save

- **Bits 2..0 – Res: Reserved Bits**

These bits are reserved bits in the ATmega103(L) and always read as zero.

XTAL Divide Control Register – XDIV

The XTAL Divide Control Register is used to divide the XTAL clock frequency by a number in the range 1 - 129. This feature can be used to decrease power consumption when the requirement for processing power is low.

Bit	7	6	5	4	3	2	1	0	
$3C ($5C)	XDIVEN	XDIV6	XDIV5	XDIV4	XDIV3	XDIV2	XDIV1	XDIV0	XDIV
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 – XDIVEN: XTAL Divide Enable**

When the XDIVEN bit is set (one), the clock frequency of the CPU and all peripherals is divided by the factor defined by the setting of XDIV6 - XDIV0. This bit can be set and cleared run-time to vary the clock frequency as suitable to the application.

- **Bits 6..0 – XDIV6..XDIV0: XTAL Divide Select Bits 6 - 0**

These bits define the division factor that applies when the XDIVEN bit is set (one). If the value of these bits is denoted *d*, the following formula defines the resulting CPU clock frequency f_{clk}:

$$f_{CLK} = \frac{XTAL}{129 - d}$$

The value of these bits can only be changed when XDIVEN is zero. When XDIVEN is set to one, the value written simultaneously into XDIV6..XDIV0 is taken as the division factor. When XDIVEN is cleared to zero, the value written simultaneously into XDIV6..XDIV0 is rejected. As the divider divides the Master Clock Input to the MCU, the speed of all peripherals is reduced when a division factor is used.

Reset and Interrupt Handling

The ATmega103(L) provides 23 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 that 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 4. The list also determines the priority levels of the different interrupts. The lower the address, the

higher the priority level. RESET has the highest priority and next is INT0 (the External Interrupt Request 0), etc.

Table 4. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$0000	RESET	Hardware Pin, Power-on Reset and Watchdog Reset
2	$0002	INT0	External Interrupt Request 0
3	$0004	INT1	External Interrupt Request 1
4	$0006	INT2	External Interrupt Request 2
5	$0008	INT3	External Interrupt Request 3
6	$000A	INT4	External Interrupt Request 4
7	$000C	INT5	External Interrupt Request 5
8	$000E	INT6	External Interrupt Request 6
9	$0010	INT7	External Interrupt Request 7
10	$0012	TIMER2 COMP	Timer/Counter2 Compare Match
11	$0014	TIMER2 OVF	Timer/Counter2 Overflow
12	$0016	TIMER1 CAPT	Timer/Counter1 Capture Event
13	$0018	TIMER1 COMPA	Timer/Counter1 Compare Match A
14	$001A	TIMER1 COMPB	Timer/Counter1 Compare Match B
15	$001C	TIMER1 OVF	Timer/Counter1 Overflow
16	$001E	TIMER0 COMP	Timer/Counter0 Compare Match
17	$0020	TIMER0 OVF	Timer/Counter0 Overflow
18	$0022	SPI, STC	SPI Serial Transfer Complete
19	$0024	UART, RX	UART, Rx Complete
20	$0026	UART, UDRE	UART Data Register Empty
21	$0028	UART, TX	UART, Tx Complete
22	$002A	ADC	ADC Conversion Complete
23	$002C	EE READY	EEPROM Ready
24	$002E	ANALOG COMP	Analog Comparator

The most typical program setup for the Reset and Interrupt vector addresses are:

```
Address Labels  Code                    Comments
$0000           jmp     RESET           ; Reset Handler
$0002           jmp     EXT_INT0        ; IRQ0 Handler
$0004           jmp     EXT_INT1        ; IRQ1 Handler
$0006           jmp     EXT_INT2        ; IRQ2 Handler
$0008           jmp     EXT_INT3        ; IRQ3 Handler
$000A           jmp     EXT_INT4        ; IRQ4 Handler
$000C           jmp     EXT_INT5        ; IRQ5 Handler
$000E           jmp     EXT_INT6        ; IRQ6 Handler
$0010           jmp     EXT_INT7        ; IRQ7 Handler
```

```
$0012          jmp     TIM2_COMP        ; Timer2 Compare Handler
$0014          jmp     TIM2_OVF         ; Timer2 Overflow Handler
$0016          jmp     TIM1_CAPT        ; Timer1 Capture Handler
$0018          jmp     TIM1_COMPA       ; Timer1 Compare A Handler
$001A          jmp     TIM1_COMPB       ; Timer1 Compare B Handler
$001C          jmp     TIM1_OVF         ; Timer1 Overflow Handler
$001E          jmp     TIM0_COMP        ; Timer0 Compare Handler
$0020          jmp     TIM0_OVF         ; Timer0 Overflow Handler
$0022          jmp     SPI_STC          ; SPI Transfer Complete Handler
$0024          jmp     UART_RXC         ; UART RX Complete Handler
$0026          jmp     UART_DRE         ; UDR Empty Handler
$0028          jmp     UART_TXC         ; UART TX Complete Handler
$002A          jmp     ADC              ; ADC Conversion Complete Handler
$002C          jmp     EE_RDY           ; EEPROM Ready Handler
$002E          jmp     ANA_COMP         ; Analog Comparator Handler
;
$0030   MAIN:  ldi     r16, high(RAMEND); Main program start
$0031          out     SPH,r16
$0032          ldi     r16, low(RAMEND)
$0033          out     SPL,r16
$0034          <instr>  xxx
  ...    ...    ...    ...
```

Reset Sources

The ATmega103(L) has three sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the $\overline{\text{RESET}}$ pin for more than 50 ns.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.

During reset, all I/O Registers except the MCU Status Register are then set to their initial values and the program starts execution from address $0000. The instruction placed in address $0000 must be a JMP (absolute 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. The circuit diagram in Figure 23 shows the reset logic. Table 5 defines the timing and electrical parameters of the reset circuitry.