

8-bit Microcontroller with 16/32K bytes of ISP Flash and USB Controller

DATASHEET

Features

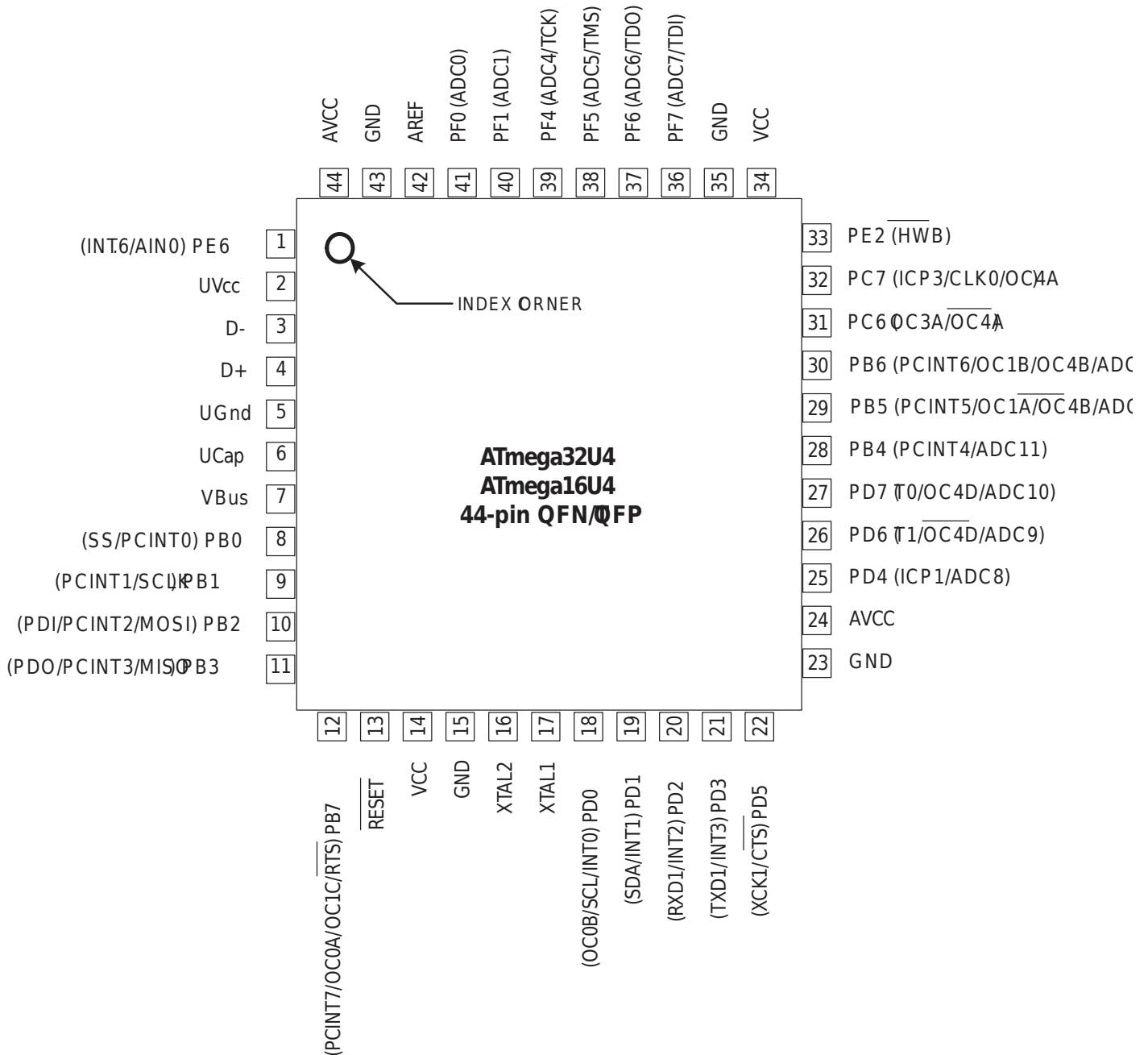
- **High Performance, Low Power AVR® 8-Bit Microcontroller**
- **Advanced RISC Architecture**
 - 135 Powerful Instructions - Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- **Non-volatile Program and Data Memories**
 - 16/32KB of In-System Self-Programmable Flash
 - 1.25/2.5KB Internal SRAM
 - 512Bytes/1KB Internal EEPROM
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Parts using external XTAL clock are pre-programmed with a default USB bootloader
 - Programming Lock for Software Security
- **J TAG (IEEE® std. 1149.1 compliant) Interface**
 - Boundary-scan Capabilities According to the J TAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the J TAG Interface
- **USB 2.0 Full-speed/Low Speed Device Module with Interrupt on Transfer Completion**
 - Complies fully with Universal Serial Bus Specification Rev 2.0
 - Supports data transfer rates up to 12Mbit/s and 1.5Mbit/s
 - Endpoint 0 for Control Transfers: up to 64-bytes
 - Six Programmable Endpoints with IN or Out Directions and with Bulk, Interrupt or Isochronous Transfers
 - Configurable Endpoints size up to 256 bytes in double bank mode
 - Fully independent 832 bytes USB DPRAM for endpoint memory allocation
 - Suspend/Resume Interrupts
 - CPU Reset possible on USB Bus Reset detection
 - 48MHz from PLL for Full-speed Bus Operation
 - USB Bus Connection/Disconnection on Microcontroller Request
 - Crystal-less operation for Low Speed mode
- **Peripheral Features**
 - On-chip PLL for USB and High Speed Timer: 32 up to 96MHz operation
 - One 8-bit Timer/Counter with Separate Prescaler and Compare Mode

- Two 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
- One 10-bit High-Speed Timer/Counter with PLL (64MHz) and Compare Mode
- Four 8-bit PWM Channels
- Four PWM Channels with Programmable Resolution from 2 to 16 Bits
- Six PWM Channels for High Speed Operation, with Programmable Resolution from 2 to 11 Bits
- Output Compare Modulator
- 12-channels, 10-bit ADC (features Differential Channels with Programmable Gain)
- Programmable Serial USART with Hardware Flow Control
- Master/Slave SPI Serial Interface
- Byte Oriented 2-wire Serial Interface
- Programmable Watchdog Timer with Separate On-chip Oscillator
- On-chip Analog Comparator
- Interrupt and Wake-up on Pin Change
- On-chip Temperature Sensor
- **Special Microcontroller Features**
 - Power-on Reset and Programmable Brown-out Detection
 - Internal 8MHz Calibrated Oscillator
 - Internal clock prescaler and On-the-fly Clock Switching (Int RC / Ext Osc)
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- **I/O and Packages**
 - All I/O combine CMOS outputs and LVTTTL inputs
 - 26 Programmable I/O Lines
 - 44-lead TQFP Package, 10x10mm
 - 44-lead QFN Package, 7x7mm
- **Operating Voltages**
 - 2.7 - 5.5V
- **Operating temperature**
 - Industrial (-40°C to +85°C)
- **Maximum Frequency**
 - 8MHz at 2.7V - Industrial range
 - 16MHz at 4.5V - Industrial range

Note: 1. See [“Data Retention” on page 8](#) for details.

1. Pin Configurations

Figure 1-1. Pinout

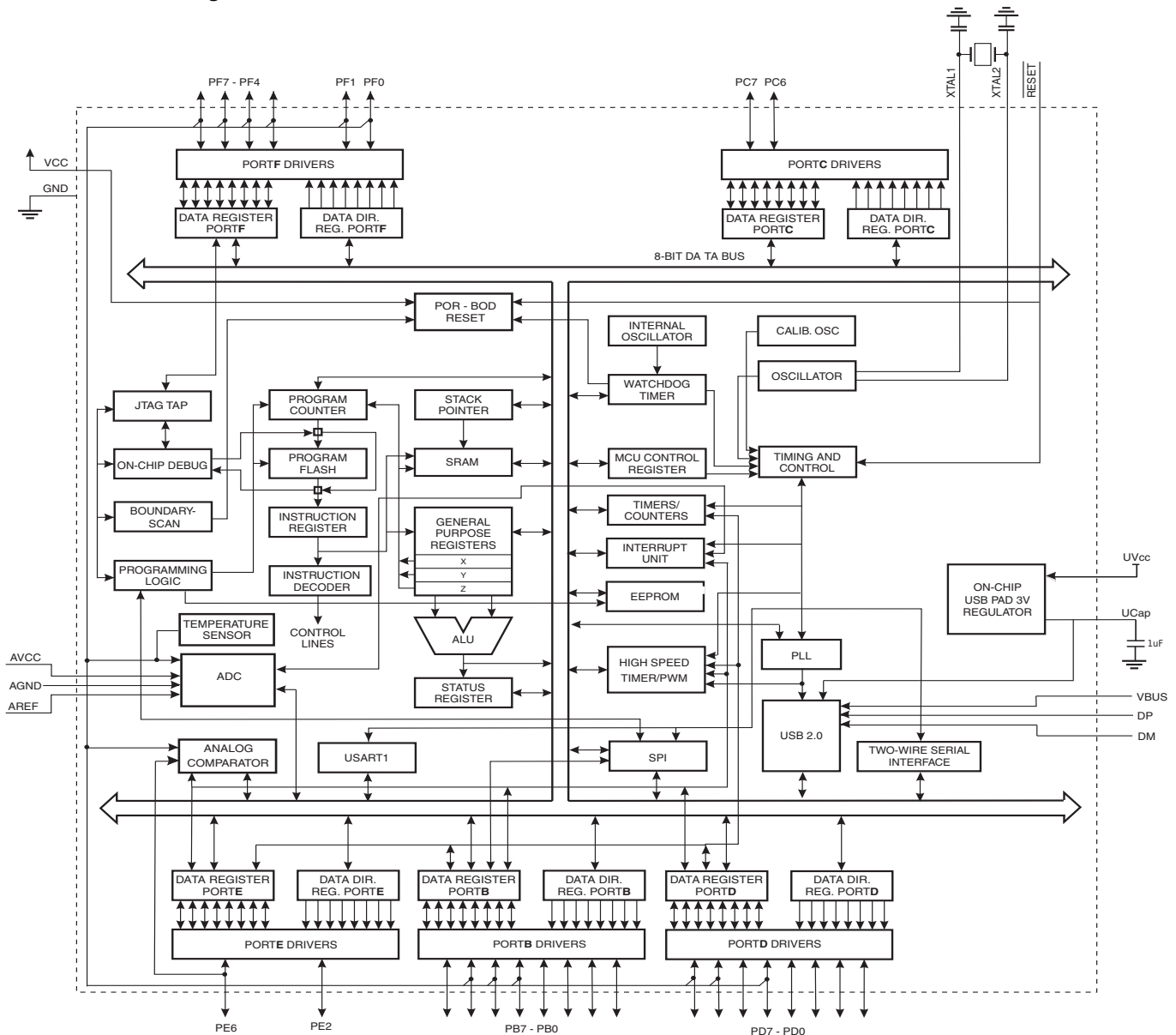


2. Overview

The [ATmega16U4/ATmega32U4](#) 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 device achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core 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 device provides the following features: 16/32K bytes of In-System Programmable Flash with Read-While-Write capabilities, 512Bytes/1K bytes EEPROM, 1.25/2.5K bytes SRAM, 26 general purpose I/O lines (CMOS outputs and LVTTTL inputs), 32 general purpose working registers, four flexible Timer/Counters with compare modes and PWM, one more high-speed Timer/Counter with compare modes and PLL adjustable source, one USART (including CTS/RTS flow control signals), a byte oriented 2-wire Serial Interface, a 12-channels 10-bit ADC with optional differential input stage with programmable gain, an on-chip calibrated temperature sensor, a programmable Watchdog Timer with Internal Oscillator, an SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and six 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 ADC Noise Reduction mode stops the CPU and all I/O modules except ADC, to minimize switching noise during ADC conversions. In Standby mode, the Crystal/Resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using the Atmel® high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system through an SPI serial interface, by a conventional nonvolatile memory programmer, or by an On-chip Boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the device is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The [ATmega16U4/ATmega32U4](#) 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.

2.2 Pin Descriptions

2.2.1 VCC

Digital supply voltage.

2.2.2 GND

Ground.

2.2.3 Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the device as listed on page 74.

2.2.4 Port C (PC7,PC6)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Only bits 6 and 7 are present on the product pinout.

Port C also serves the functions of special features of the device as listed on page 77.

2.2.5 Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the [ATmega16U4/ATmega32U4](#) as listed on page 78.

2.2.6 Port E (PE6,PE2)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Only bits 2 and 6 are present on the product pinout.

Port E also serves the functions of various special features of the [ATmega16U4/ATmega32U4](#) as listed on page 81.

2.2.7 Port F (PF7..PF4, PF1,PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter channels are not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Bits 2 and 3 are not present on the product pinout.

Port F also serves the functions of the JTAG interface. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

2.2.8 D-

USB Full speed / Low Speed Negative Data Upstream Port. Should be connected to the USB D- connector pin with a serial 22Ω resistor.

2.2.9 D+

USB Full speed / Low Speed Positive Data Upstream Port. Should be connected to the USB D+ connector pin with a serial 22Ω resistor.

2.2.10 UGND

USB Pads Ground.

2.2.11 UVCC

USB Pads Internal Regulator Input supply voltage.

2.2.12 UCAP

USB Pads Internal Regulator Output supply voltage. Should be connected to an external capacitor (1μF).

2.2.13 VBUS

USB VBUS monitor input.

2.2.14 RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in [Table 8-2 on page 53](#). Shorter pulses are not guaranteed to generate a reset.

2.2.15 XTAL1

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

2.2.16 XTAL2

Output from the inverting Oscillator amplifier.

2.2.17 AVCC

AVCC is the supply voltage pin (input) for all the A/D Converter channels. If the ADC is not used, it should be externally connected to V_{CC} . If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

2.2.18 AREF

This is the analog reference pin (input) for the A/D Converter.

3. About

3.1 Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min. and Max. values will be available after the device is characterized.

3.2 Resources

A comprehensive set of development tools, application notes and datasheets are available for download on <http://www.atmel.com/avr>.

3.3 Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Confirm with the C compiler documentation for more details.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBR", "SBRC", "SBR", and "CBR".

3.4 Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1PPM over 20 years at 85°C or 100 years at 25°C.

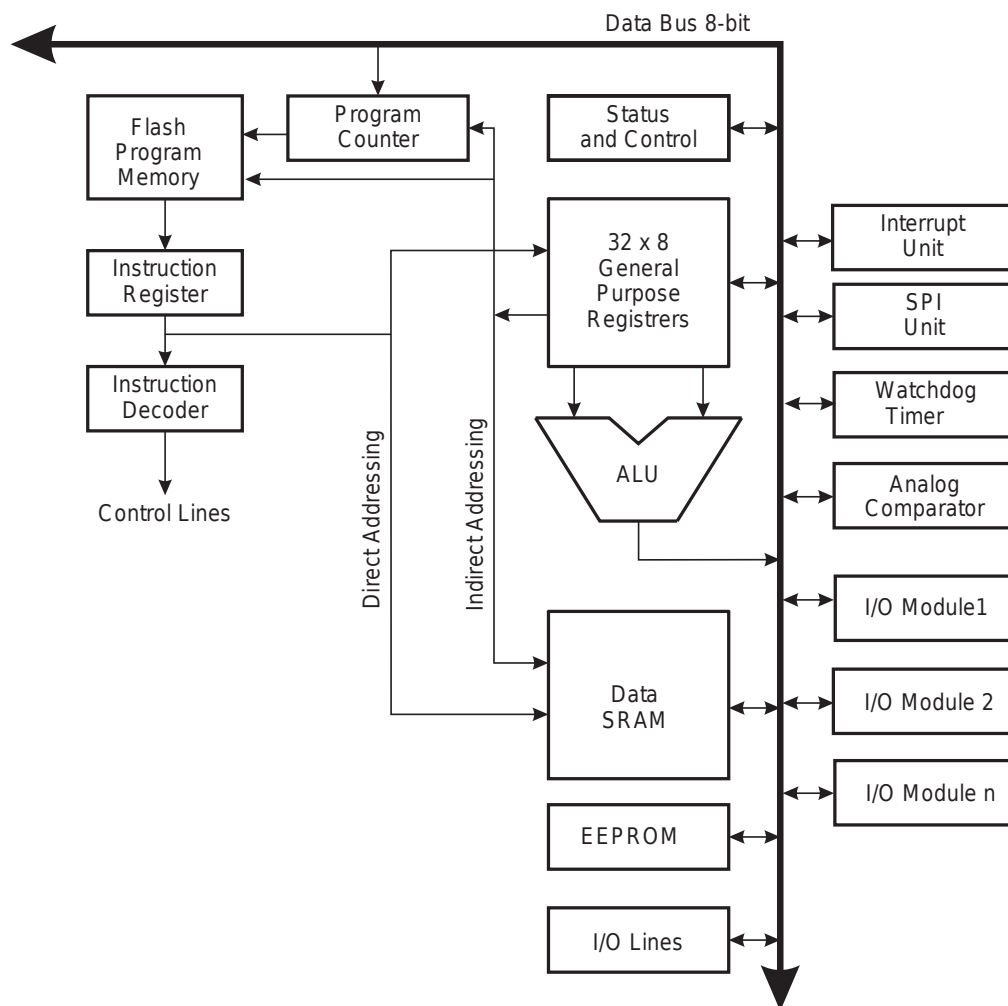
4. AVR CPU Core

4.1 Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculations, control peripherals, and handle interrupts.

4.2 Architectural Overview

Figure 4-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are 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 Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two

operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

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

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

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 Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

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

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the [ATmega16U4/ATmega32U4](#) has Extended I/O space from 0x60 - 0x0FF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

4.3 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, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See [“Instruction Set Summary” on page 418](#) for a detailed description.

4.4 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
	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 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, 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. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 - T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit LoaD) and BST (Bit STore) use the T-bit as source or 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. Half Carry Is useful in BCD arithmetic. See [“Instruction Set Summary” on page 418](#) 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 [“Instruction Set Summary” on page 418](#) for detailed information.

- **Bit 3 - V: Two’s Complement Overflow Flag**

The Two’s Complement Overflow Flag V supports two’s arithmetic complements. See [“Instruction Set Summary” on page 418](#) for detailed information.

- **Bit 2 - N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See [“Instruction Set Summary” on page 418](#) for detailed information.

- **Bit 1 - Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See [“Instruction Set Summary” on page 418](#) for detailed information.

- **Bit 0 - C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See [“Instruction Set Summary” on page 418](#) for detailed information.

4.5 General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input

- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 4-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 4-2. AVR CPU General Purpose Working Registers

	7	0	Addr.	
General Purpose Working Registers	R0		0x00	
	R1		0x01	
	R2		0x02	
	...			
	R13		0x0D	
	R14		0x0E	
	R15		0x0F	
	R16		0x10	
	R17		0x11	
	...			
	R26		0x1A	X-register Low Byte
	R27		0x1B	X-register High Byte
	R28		0x1C	Y-register Low Byte
	R29		0x1D	Y-register High Byte
	R30		0x1E	Z-register Low Byte
	R31		0x1F	Z-register High Byte

Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 4-2, 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-pointer registers can be set to index any register in the file.

4.5.1 The X-register, Y-register, and Z-register

The registers R26..R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 4-3.