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

ATmega328PB

AVR® Microcontroller with Core Independent Peripherals and PicoPower® Technology

Introduction

The picoPower® ATmega328PB 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 ATmega328PB achieves throughputs close to 1 MIPS per MHz. This empowers system designers to optimize the device for power consumption versus processing speed.

Features

High Performance, Low-Power AVR® 8-bit Microcontroller Family

- Advanced RISC Architecture
  - 131 Powerful Instructions
  - Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-Chip 2-Cycle Multiplier
- High Endurance Nonvolatile Memory Segments
  - 32 KB of In-System Self-Programmable Flash program memory
  - 1 KB EEPROM
  - 2 KB Internal SRAM
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Optional Boot Code Section with Independent Lock Bits
    - In-System Programming by On-chip Boot Program
    - True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Peripheral Touch Controller (PTC)
    - Capacitive Touch Buttons, Sliders, and Wheels
    - 24 Self-Cap Channels and 144 Mutual Cap Channels
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - Three 16-bit Timer/Counters with Separate Prescaler, Compare Mode, and Capture Mode
  - Real-Time Counter with Separate Oscillator
  - Ten PWM Channels
  - 8-channel 10-bit ADC

  - Two Programmable Serial USARTs
  - Two Master/Slave SPI Serial Interfaces
  - Two Byte-Oriented Two-Wire Serial Interfaces (Philips I^2C Compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-Chip Analog Comparator
  - Interrupt and Wake-Up on Pin Change
- Special Microcontroller Features
  - Power-On Reset and Programmable Brown-Out Detection
  - Internal 8 MHz Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
  - Clock Failure Detection Mechanism and Switch to Internal 8 MHz RC Oscillator in case of Failure
  - Individual Serial Number to Represent a Unique ID
- I/O and Packages
  - 27 Programmable I/O Lines
  - 32-pin TQFP and 32-pin QFN /MLF
- Operating Voltage:
  - 1.8 - 5.5V
- Temperature Range:
  - -40°C to 105°C
- Speed Grade:
  - 0 - 4 MHz @ 1.8 - 5.5V
  - 0 - 10 MHz @ 2.7 - 5.5V
  - 0 - 20 MHz @ 4.5 - 5.5V
- Power Consumption at 1 MHz, 1.8V, 25°C
  - Active Mode: 0.24 mA
  - Power-Down Mode: 0.2 µA
  - Power-Save Mode: 1.3 µA (Including 32 kHz RTC)

Table of Contents

1. Description

The ATmega328PB 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 ATmega328PB achieves throughputs close to 1 MIPS per MHz. This empowers system designer to optimize the device for power consumption versus processing speed.

The 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 a 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 ATmega328PB provides the following features: 32 KB of In-System Programmable Flash with Read-While-Write capabilities, 1 KB EEPROM, 2 KB SRAM, 27 general purpose I/O lines, 32 general purpose working registers, five flexible Timer/Counters with compare modes, internal and external interrupts, two serial programmable USART, two byte-oriented two-wire Serial Interface (I^2C), two SPI serial ports, an 8-channel 10-bit ADC in TQFP and QFN/MLF package, a programmable Watchdog Timer with internal Oscillator, Clock failure detection mechanism, and six software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, two-wire Serial Interface, SPI port, and interrupt system to continue functioning. PTC with enabling up to 24 self-cap and 144 mutual-cap sensors. 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 asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. Also ability to run PTC in Power-Save mode/wake-up on touch and Dynamic ON/OFF of PTC analog and digital portion. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer, PTC, and 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 high-density non-volatile 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 ATmega328PB is a powerful microcontroller that provides a highly flexible and cost-effective solution to many embedded control applications.

The ATmega328PB is supported by a full suite of program and system development tools including C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.

2. Configuration Summary

Features	ATmega328PB
Pin count	32
Flash (KB)	32
SRAM (KB)	2
EEPROM (KB)	1
General Purpose I/O pins	27
SPI	2
TWI (I^2C)	2
USART	2
ADC	10-bit 15 ksps
ADC channels	8
AC propagation delay	400 ns (Typical)
8-bit Timer/Counters	2
16-bit Timer/Counters	3
PWM channels	10
PTC	Available
Clock Failure Detector (CFD)	Available
Output Compare Modulator (OCM1C2)	Available

3. Ordering Information

Speed [MHz]	Power Supply [V]	Ordering Code[2]	Package[1]	Operational Range
20	1.8 - 5.5	ATmega328PB-AU ATmega328PB-AUR[3] ATmega328PB-MU ATmega328PB-MUR[3]	32A 32A 32MS1 32MS1	Industrial (-40°C to 85°C)
		ATmega328PB-AN ATmega328PB-ANR[3] ATmega328PB-MN ATmega328PB-MNR[3]	32A 32A 32MS1 32MS1	Industrial (-40°C to 105°C)

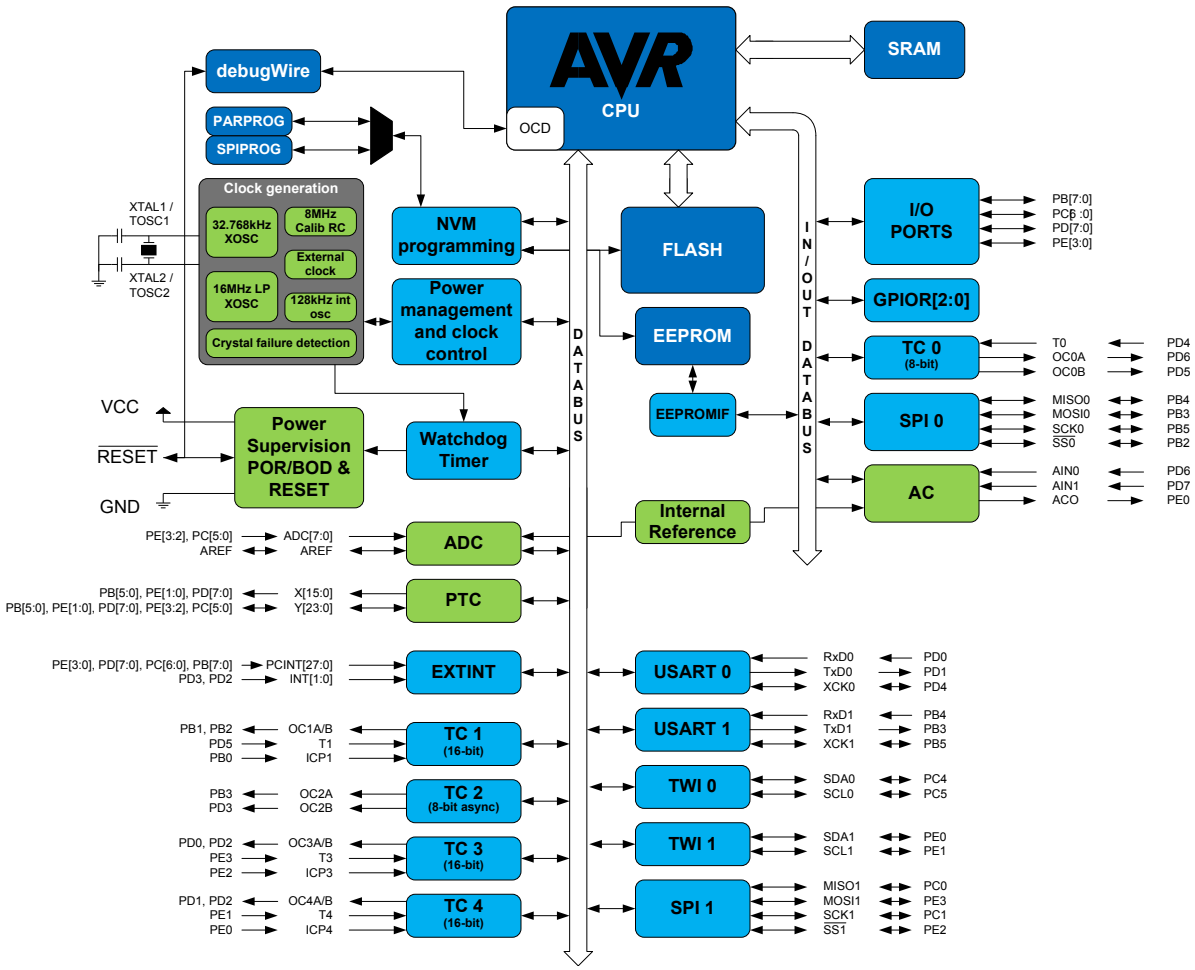
Note:

1. This device can also be supplied in wafer form. Contact your local Microchip sales office for detailed ordering information and minimum quantities.
2. Pb-free packaging complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
3. Tape & Reel.

Package Type	
32A	32-lead, Thin (1.0 mm) Plastic Quad Flat Package (TQFP)
32MS1	32-pad, 5.0x5.0x0.9 mm body, Lead Pitch 0.50 mm, Very-thin Fine pitch, Quad Flat No Lead Package (VQFN)

4. Block Diagram

Figure 4-1. Block Diagram



5. Pin Configurations

Figure 5-1. 32 TQFP Pinout ATmega328PB

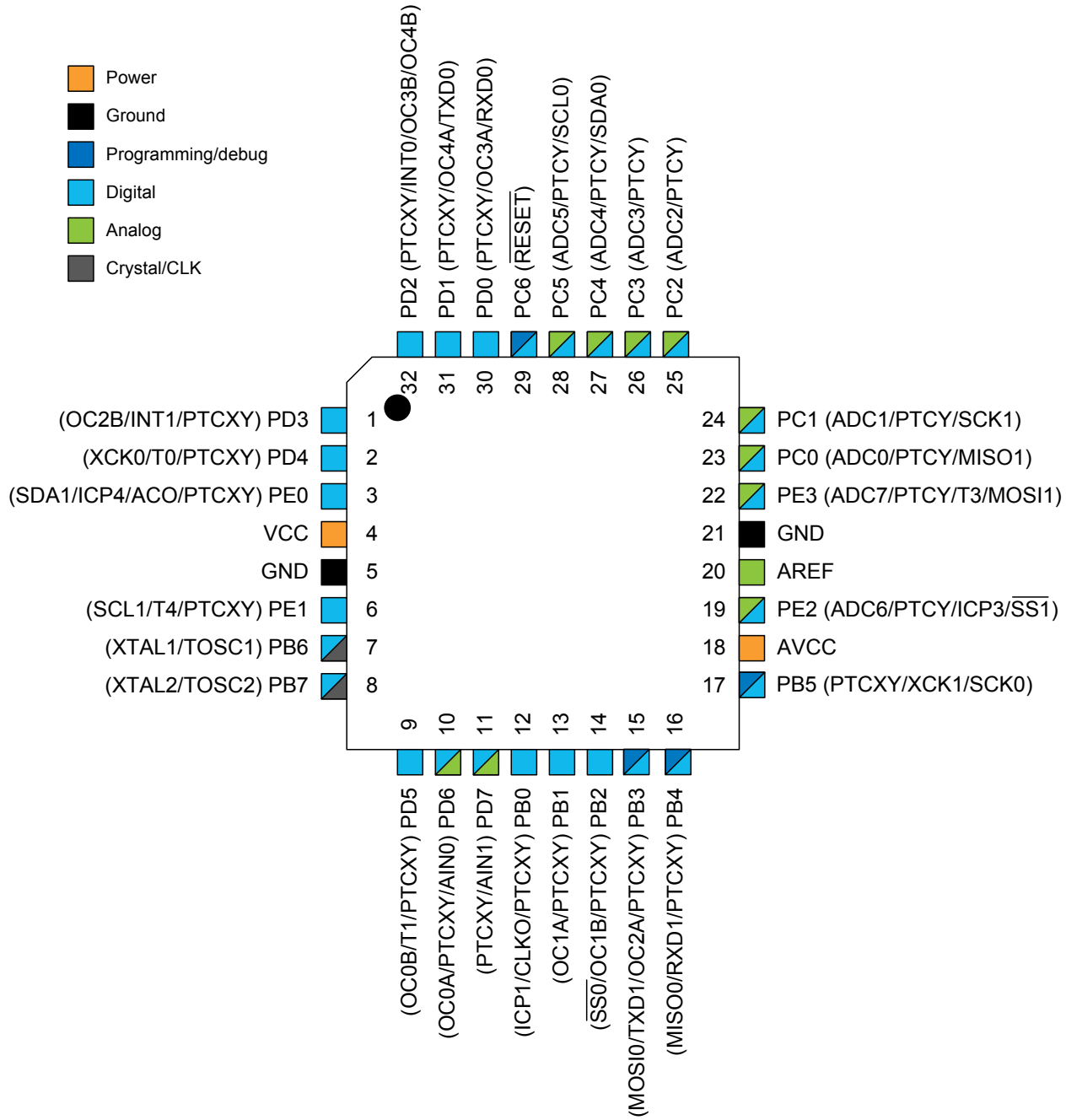
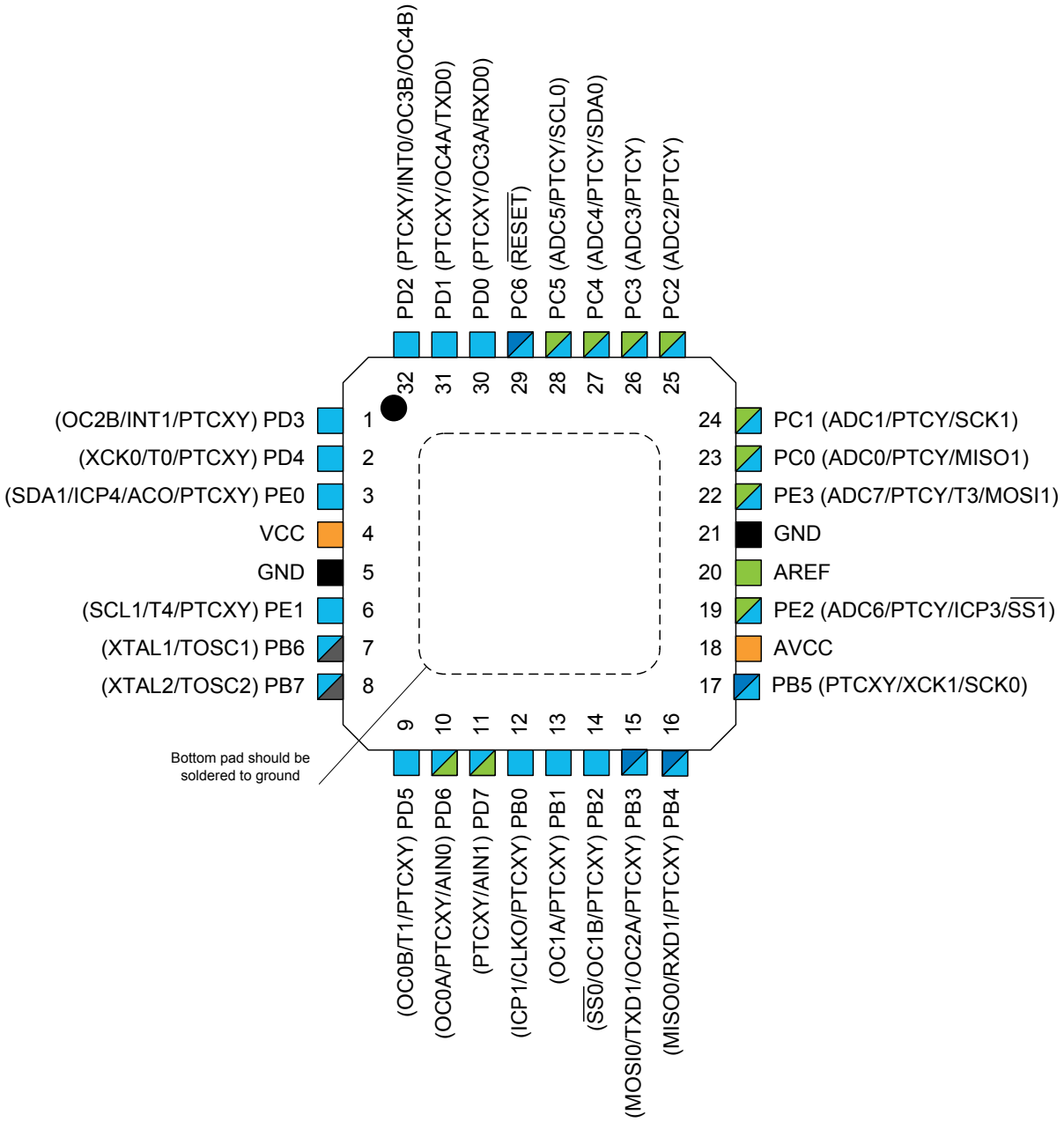


Figure 5-2. 32 VQFN Pinout ATmega328PB



5.1 Pin Descriptions

5.1.1 VCC

Digital supply voltage pin.

5.1.2 GND

Ground.

5.1.3 Port B (PB[7:0]) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each pin). 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 during a reset condition even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB[7:6] is used as TOSC[2:1] input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

5.1.4 Port C (PC[5:0])

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each pin). The PC[5:0] 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 during a reset condition even if the clock is not running.

5.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a 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. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in the *Alternate Functions of Port C* section.

5.1.6 Port D (PD[7:0])

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each pin). 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 during a reset condition even if the clock is not running.

5.1.7 Port E (PE[3:0])

Port E is a 4-bit bi-directional I/O port with internal pull-up resistors (selected for each pin). 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 during a reset condition even if the clock is not running.

5.1.8 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC[3:0], and PE[3:2]. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC[6:4] use digital supply voltage, V_{CC}.

5.1.9 AREF

AREF is the analog reference pin for the A/D Converter.

5.1.10 ADC[7:6]

In the TQFP and VFQFN package, ADC[7:6] serve as analog inputs to the A/D converter. These pins are powered by the analog supply and serve as 10-bit ADC channels.

6. I/O Multiplexing

Each pin is by default controlled by the PORT as a general purpose I/O and alternatively, it can be assigned to one of the peripheral functions.

The following table describes the peripheral signals multiplexed to the PORT I/O pins.

Table 6-1. PORT Function Multiplexing

No	PAD	EXTINT	PCINT	ADC/AC	PTC X	PTC Y	OSC	T/C	USART	I2C	SPI
1	PD[3]	INT1	PCINT19		X3	Y11		OC2B			
2	PD[4]		PCINT20		X4	Y12		T0	XCK0		
3	PE[0]		PCINT24	ACO	X8	Y16		ICP4		SDA1	
4	VCC										
5	GND										
6	PE[1]		PCINT25		X9	Y17		T4		SCL1	
7	PB[6]		PCINT6				XTAL1/TOSC1				
8	PB[7]		PCINT7				XTAL2/TOSC2				
9	PD[5]		PCINT21		X5	Y13		OC0B / T1			
10	PD[6]		PCINT22	AIN0	X6	Y14		OC0A			
11	PD[7]		PCINT23	AIN1	X7	Y15					
12	PB[0]		PCINT0		X10	Y18	CLKO	ICP1			
13	PB[1]		PCINT1		X11	Y19		OC1A			
14	PB[2]		PCINT2		X12	Y20		OC1B			$\overline{SS0}$
15	PB[3]		PCINT3		X13	Y21		OC2A	TXD1		MOSI0
16	PB[4]		PCINT4		X14	Y22			RXD1		MISO0
17	PB[5]		PCINT5		X15	Y23			XCK1		SCK0
18	AVCC										
19	PE[2]		PCINT26	ADC6		Y6		ICP3			$\overline{SS1}$
20	AREF										
21	GND										
22	PE[3]		PCINT27	ADC7		Y7		T3			MOSI1
23	PC[0]		PCINT8	ADC0		Y0					MISO1
24	PC[1]		PCINT9	ADC1		Y1					SCK1
25	PC[2]		PCINT10	ADC2		Y2					
26	PC[3]		PCINT11	ADC3		Y3					
27	PC[4]		PCINT12	ADC4		Y4				SDA0	
28	PC[5]		PCINT13	ADC5		Y5				SCL0	
29	PC[6]/$\overline{RESET}$		PCINT14								
30	PD[0]		PCINT16		X0	Y8		OC3A	RXD0		

No	PAD	EXTINT	PCINT	ADC/AC	PTC X	PTC Y	OSC	T/C	USART	I2C	SPI
31	PD[1]		PCINT17		X1	Y9		OC4A	TXD0		
32	PD[2]	INT0	PCINT18		X2	Y10		OC3B / OC4B			

7. Resources

A comprehensive set of development tools, application notes, and datasheets are available for download on http://www.microchip.com/design-centers/8-bit/microchip-avr-mcus.

8. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. 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.

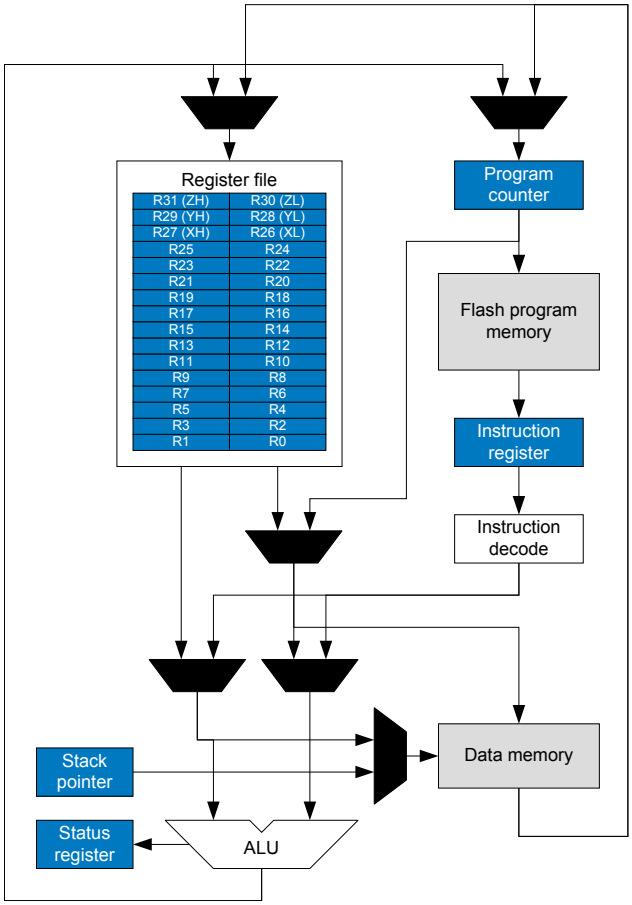
For I/O Registers located in extended I/O map, “IN”, “OUT”, “SBIS”, “SBIC”, “CBI”, and “SBI” instructions must be replaced with instructions that allow access to extended I/O. Typically “LDS” and “STS” combined with “SBRS”, “SBRC”, “SBR”, and “CBR”.

9. AVR CPU Core

9.1 Overview

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.

Figure 9-1. 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 be used as an

address pointer for lookup 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 into 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 Stack Pointer (SP) in the Reset routine (before subroutines or interrupts are executed). The 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, this device has extended I/O space from 0x60 - 0xFF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

9.2 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* section for a detailed description.

Related Links

Instruction Set Summary

9.3 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. 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.

9.3.1 Status Register

Name: SREG
Offset: 0x5F
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x3F

When addressing I/O registers as data space using LD and ST instructions, the provided offset must be used. When using the I/O specific commands IN and OUT, the offset is reduced by 0x20, resulting in an I/O address offset within 0x00 - 0x3F.

The device is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in Opcode for the IN and OUT instructions. For the extended I/O space from 0x60 in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

Bit	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	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 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 flag is useful in BCD arithmetic. See the *Instruction Set Description* for detailed information.

Bit 4 – S Sign Flag, 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 arithmetic. See the *Instruction Set Description* for detailed information.

Bit 2 – N Negative Flag
The negative flag N indicates a negative result in an arithmetic or logic operation. See the *Instruction Set Description* for detailed information.