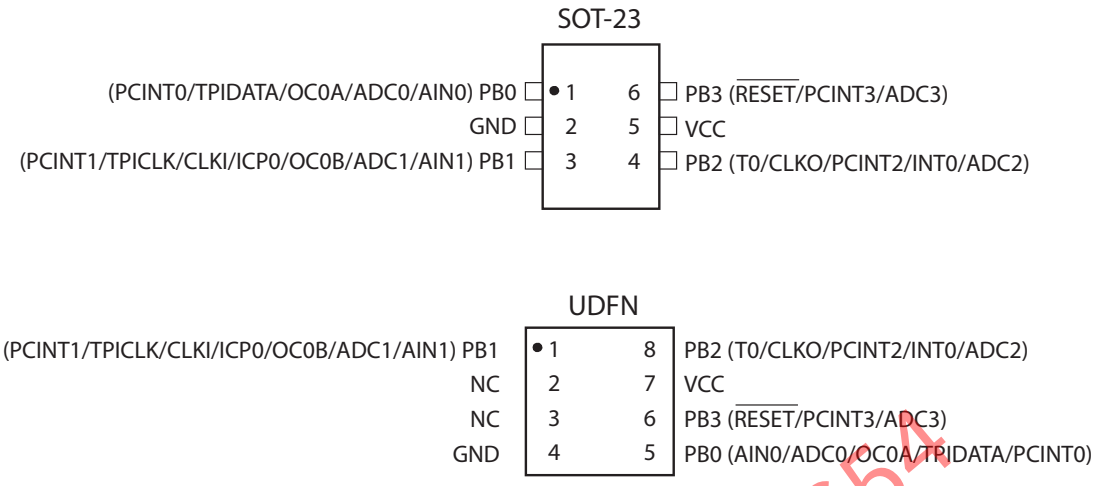
1. Pin Configurations

Figure 1-1. Pinout of ATtiny4/5/9/10



1.1 Pin Description

1.1.1 VCC

Supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB3..PB0)

This is a 4-bit, bi-directional I/O port with internal pull-up resistors, individually selectable for each bit. The output buffers have symmetrical drive characteristics, with both high sink and source capability. As inputs, the port pins that are externally pulled low will source current if pull-up resistors are activated. Port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The port also serves the functions of various special features of the ATtiny4/5/9/10

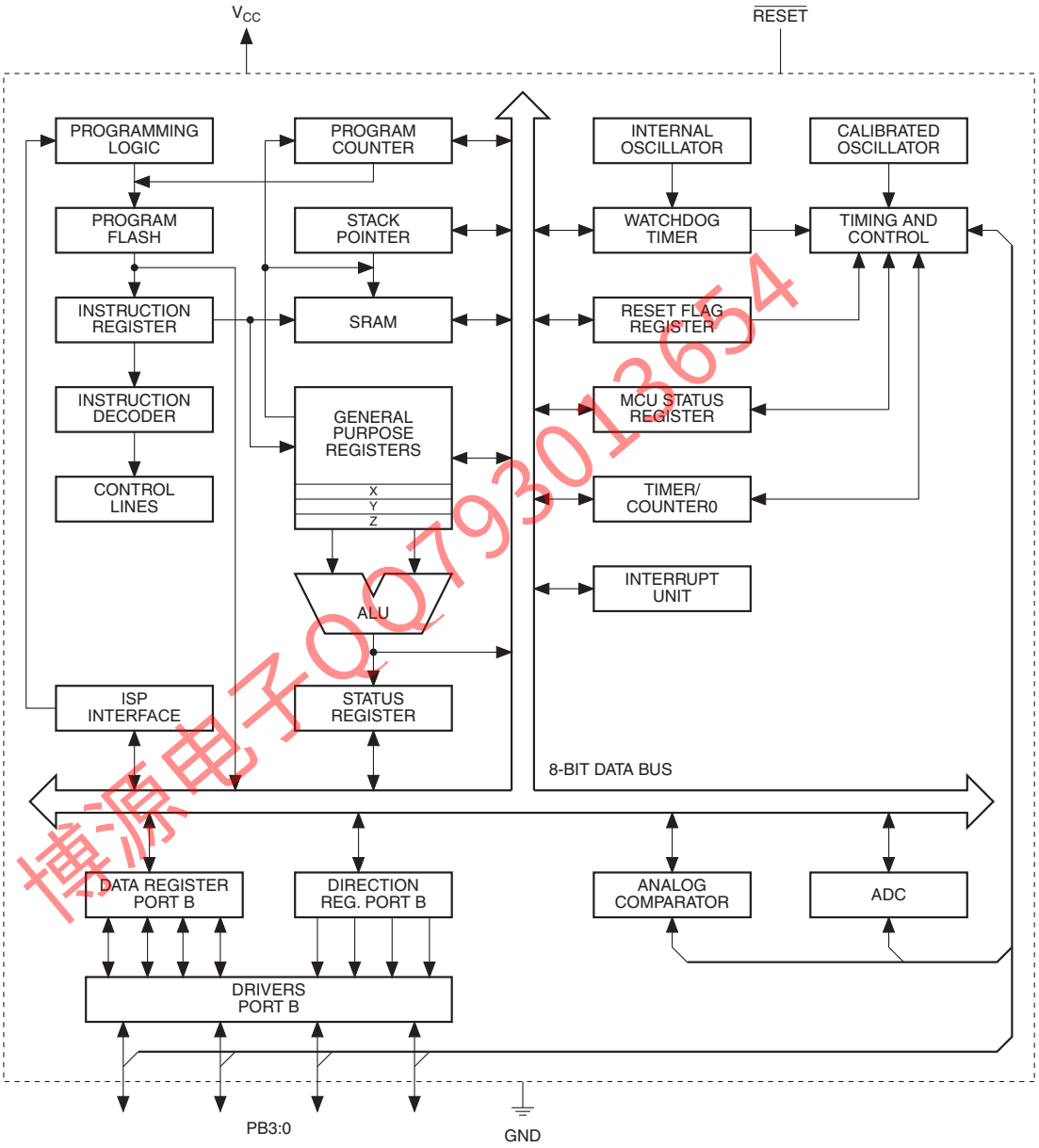
1.1.4 $\overline{\text{RESET}}$

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock

2. Overview

ATtiny4/5/9/10 are low-power CMOS 8-bit microcontrollers based on the compact AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny4/5/9/10 achieve throughputs approaching 1 MIPS per MHz, allowing the system designer to optimize power consumption versus processing speed.

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 16 general purpose working registers and system registers. All 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 compact and code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

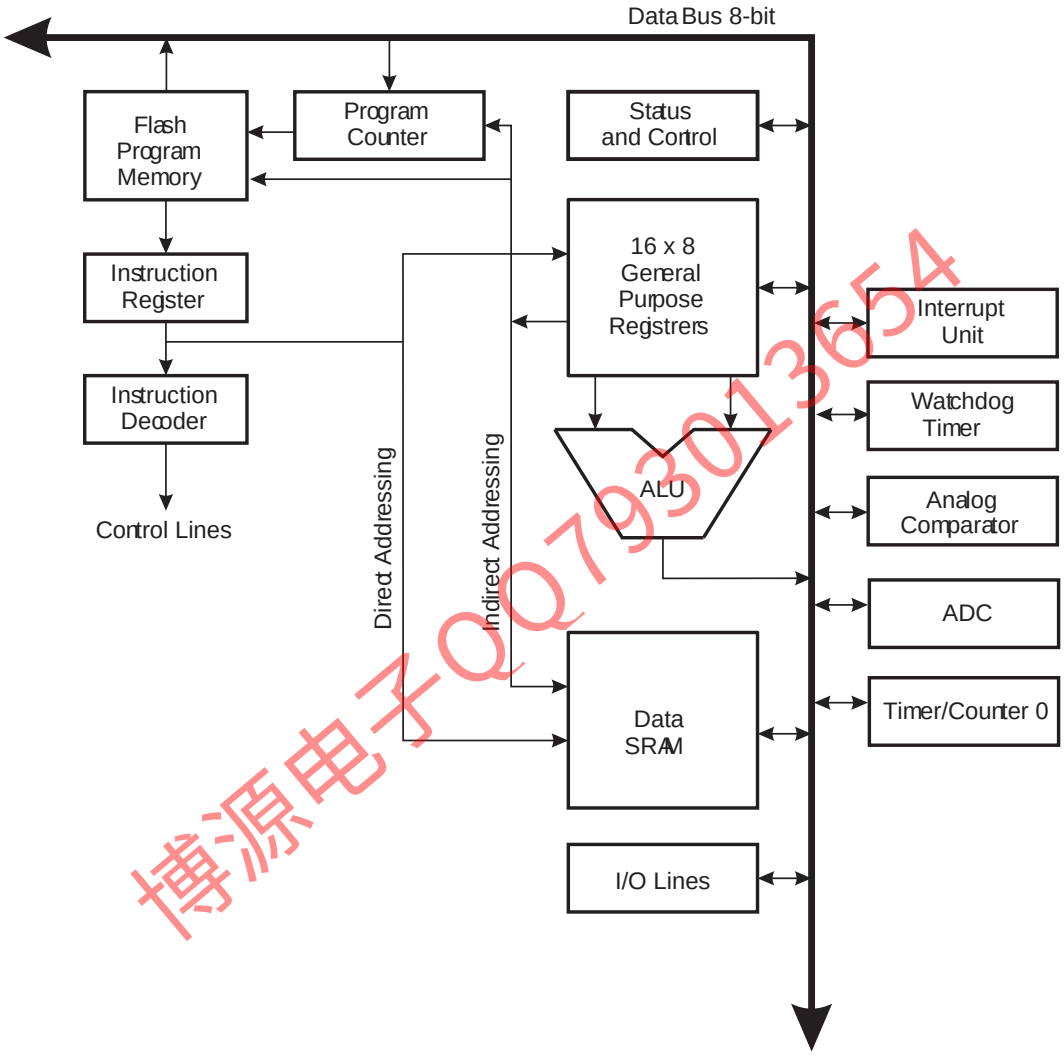
The ATtiny4/5/9/10 provide the following features: 512/1024 byte of In-System Programmable Flash, 32 bytes of SRAM, four general purpose I/O lines, 16 general purpose working registers, a 16-bit timer/counter with two PWM

4. CPU Core

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.

4.1 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 16 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.

Figure 5-1. Data Memory Map (Byte Addressing)

I/O SPACE	0x0000 ... 0x003F
SRAM DATA MEMORY	0x0040 ... 0x005F
(reserved)	0x0060 ... 0x3EFF
NVM LOCK BITS	0x3F00 ... 0x3F01
(reserved)	0x3F02 ... 0x3F3F
CONFIGURATION BITS	0x3F40 ... 0x3F41
(reserved)	0x3F42 ... 0x3F7F
CALIBRATION BITS	0x3F80 ... 0x3F81
(reserved)	0x3F82 ... 0x3FBF
DEVICE ID BITS	0x3FC0 ... 0x3FC3
(reserved)	0x3FC4 ... 0x3FFF
FLASH PROGRAM MEMORY	0x4000 ... 0x41FF/0x43FF
(reserved)	0x4400 ... 0xFFFF

5.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described i

Figure 5-2. On-chip Data SRAM Access Cycles