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!

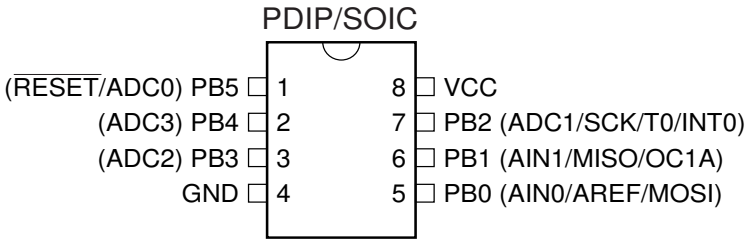
Contact us

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

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
  - 90 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
- Non-volatile Program and Data Memories
  - 1K Byte In-System Programmable Flash Program Memory
    Endurance: 1,000 Write/Erase Cycles
  - 64 Bytes EEPROM
    Endurance: 100,000 Write/Erase Cycles
  - Programming Lock for Flash Program Data Security
- Peripheral Features
  - Interrupt and Wake-up on Pin Change
  - Two 8-bit Timer/Counters with Separate Prescalers
  - One 150 kHz, 8-bit High-speed PWM Output
  - 4-channel 10-bit ADC
    One Differential Voltage Input with Optional Gain of 20x
  - On-chip Analog Comparator
  - Programmable Watchdog Timer with On-chip Oscillator
- Special Microcontroller Features
  - In-System Programmable via SPI Port
  - Enhanced Power-on Reset Circuit
  - Programmable Brown-out Detection Circuit
  - Internal, Calibrated 1.6 MHz Tunable Oscillator
  - Internal 25.6 MHz Clock Generator for Timer/Counter
  - External and Internal Interrupt Sources
  - Low-power Idle and Power-down Modes
- Power Consumption at 1.6 MHz, 3V, 25°C
  - Active: 3.0 mA
  - Idle Mode: 1.0 mA
  - Power-down: < 1 µA
- I/O and Packages
  - 8-lead PDIP and 8-lead SOIC: 6 Programmable I/O Lines
- Operating Voltages
  - 2.7V - 5.5V
- Internal 1.6 MHz System Clock

Pin Configuration



8-bit AVR® Microcontroller with 1K Byte Flash

ATtiny15L

Not recommended for new design

Rev. 1187H–AVR–09/07

Description

The ATtiny15L is a low-power CMOS 8-bit microcontroller based on the AVR RISC architecture. By executing powerful instructions in a single clock cycle, the ATtiny15L achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

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.

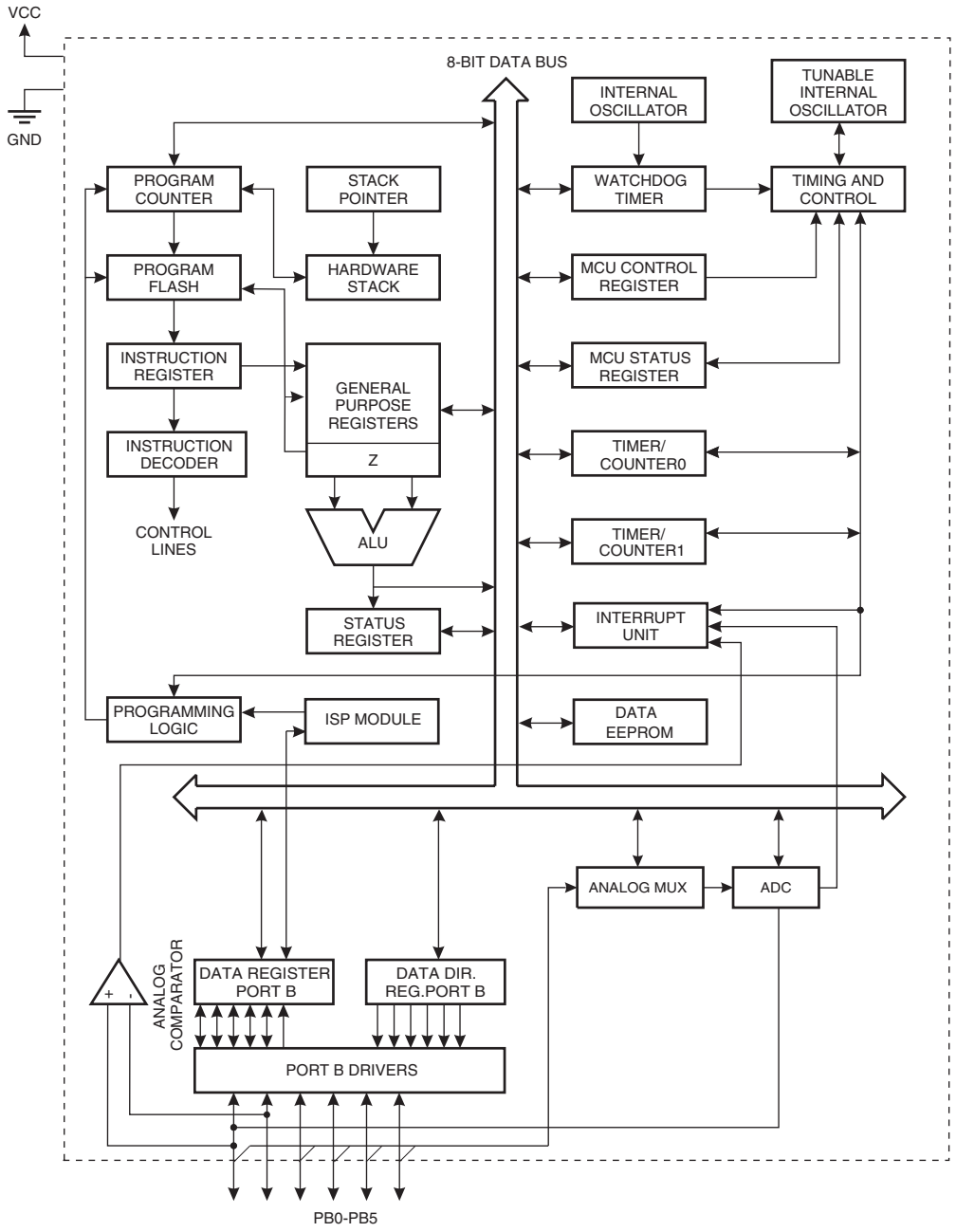
The ATtiny15L provides 1K byte of Flash, 64 bytes EEPROM, six general purpose I/O lines, 32 general purpose working registers, two 8-bit Timer/Counters, one with high-speed PWM output, internal Oscillators, internal and external interrupts, programmable Watchdog Timer, 4-channel 10-bit Analog-to-Digital Converter with one differential voltage input with optional 20x gain, and three software-selectable Power-saving modes. The Idle mode stops the CPU while allowing the ADC, anAlog Comparator, Timer/Counters and interrupt system to continue functioning. The ADC Noise Reduction mode facilitates high-accuracy ADC measurements by stopping the CPU while allowing the ADC to continue functioning. The Power-down mode saves the register contents but freezes the Oscillators, disabling all other chip functions until the next interrupt or Hardware Reset. The wake-up or interrupt on pin change features enable the ATtiny15L to be highly responsive to external events, still featuring the lowest power consumption while in the Power-saving modes.

The device is manufactured using Atmel's high-density, Non-volatile memory technology. By combining a RISC 8-bit CPU with Flash on a monolithic chip, the ATtiny15L is a powerful microcontroller that provides a highly flexible and cost-efficient solution to many embedded control applications. The peripheral features make the ATtiny15L particularly suited for battery chargers, lighting ballasts and all kinds of intelligent sensor applications.

The ATtiny15L AVR is supported with a full suite of program and system development tools including macro assemblers, program debugger/simulators, In-circuit emulators and evaluation kits.

Block Diagram

Figure 1. The ATtiny15L Block Diagram



Pin Descriptions

VCC

Supply voltage pin.

GND

Ground pin.

Port B (PB5..PB0)

Port B is a 6-bit I/O port. PB4..0 are I/O pins that can provide internal pull-ups (selected for each bit). PB5 is input or open-drain output. The use of pin PB5 is defined by a fuse and the special function associated with this pin is External Reset. The port pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also accommodates analog I/O pins. The Port B pins with alternate functions are shown in Table 1.

Table 1. Port B Alternate Functions

Port Pin	Alternate Function
PB0	MOSI (Data Input Line for Memory Downloading) AREF (ADC Voltage Reference) AIN0 (Analog Comparator Positive Input)
PB1	MISO (Data Output Line for Memory Downloading) OC1A (Timer/Counter PWM Output) AIN1 (Analog Comparator Negative Input)
PB2	SCK (Serial Clock Input for Serial Programming) INT0 (External Interrupt0 Input) ADC1 (ADC Input Channel 1) T0 (Timer/Counter0 External Counter Input)
PB3	ADC2 (ADC Input Channel 2)
PB4	ADC3 (ADC Input Channel 3)
PB5	$\overline{\text{RESET}}$ (External Reset Pin) ADC0 (ADC Input Channel 0)

Analog Pins

Up to four analog inputs can be selected as inputs to Analog-to-Digital Converter (ADC).

Internal Oscillators

The internal Oscillator provides a clock rate of nominally 1.6 MHz for the system clock (CK). Due to large initial variation (0.8 -1.6 MHz) of the internal Oscillator, a tuning capability is built in. Through an 8-bit control register – OSCCAL – the system clock rate can be tuned with less than 1% steps of the nominal clock.

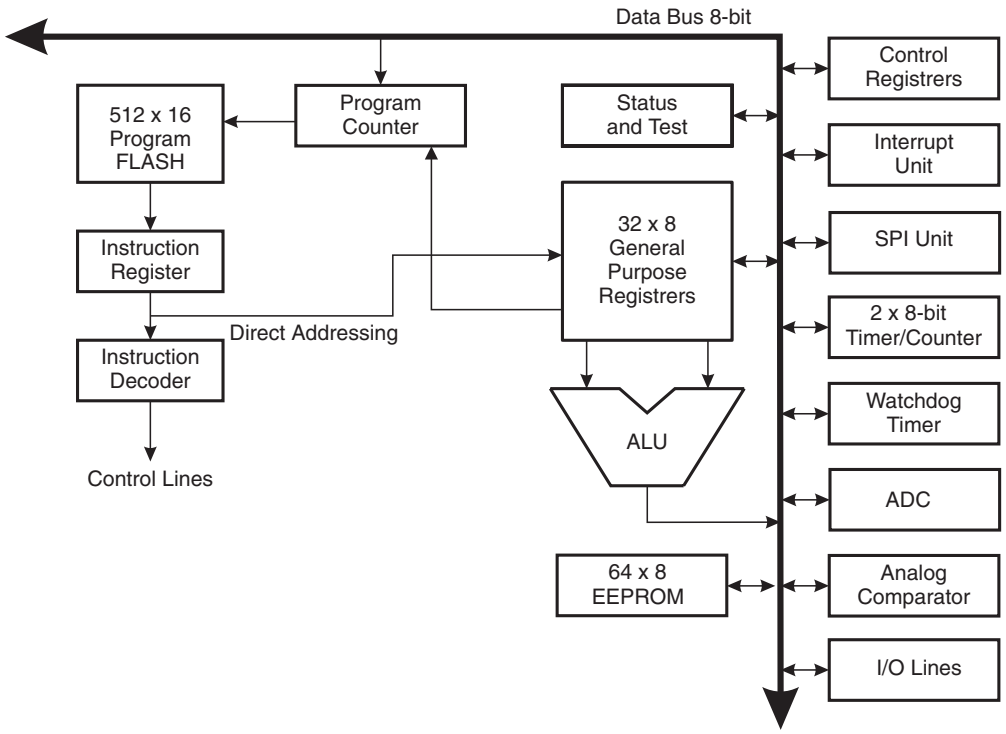
There is an internal PLL that provides a 16x clock rate locked to the system clock (CK) for the use of the Peripheral Timer/Counter1. The nominal frequency of this peripheral clock, PCK, is 25.6 MHz.

ATtiny15L Architectural Overview

The fast-access Register File concept contains 32 x 8-bit general purpose working registers with a single-clock-cycle access time. This means that during one single clock cycle, one ALU (Arithmetic Logic Unit) operation is executed. Two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Two of the 32 registers can be used as a 16-bit pointer for indirect memory access. This pointer is called the Z-pointer, and can address the Register File, IO file and the Flash Program memory.

Figure 2. The ATtiny15L AVR RISC Architecture



The ALU supports arithmetic and logic functions between registers or between a constant and a register. Single-register operations are also executed in the ALU. Figure 2 shows the ATtiny15L AVR RISC microcontroller architecture. The AVR uses a Harvard architecture concept with separate memories and buses for program and data memories. The program memory is accessed with a two-stage 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 the relative jump and relative call instructions, the whole address space is directly accessed. All AVR instructions have a single 16-bit word format, meaning that every program memory address contains a single 16-bit instruction.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is a 3-level-deep Hardware Stack dedicated for subroutines and interrupts.

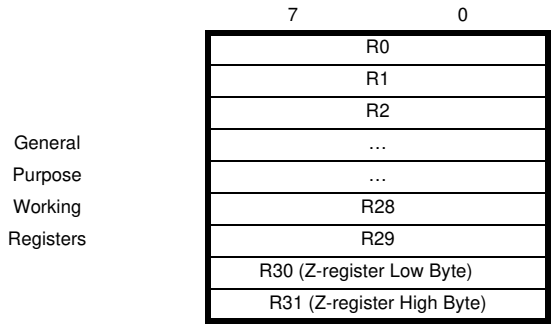
The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, Timer/Counters and other I/O functions. The memory spaces in the AVR architecture are all linear and regular memory maps.

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

The General Purpose Register File

Figure 3 shows the structure of the 32 general purpose registers in the CPU.

Figure 3. AVR CPU General Purpose Working Registers



All the register operating instructions in the instruction set have direct- and single-cycle access to all registers. The only exception is the five constant arithmetic and logic instructions SBCI, SUBI, CPI, ANDI, and ORI between a constant and a register and the LDI instruction for load-immediate constant data. These instructions apply to the second half of the registers in the Register File – R16..R31. The general SBC, SUB, CP, AND, OR, and all other operations between two registers or on a single-register apply to the entire Register File.

Registers 30 and 31 form a 16-bit pointer (the Z-pointer) which is used for indirect Flash memory and Register File access. When the Register File is accessed, the contents of R31 is discarded by the CPU.

The ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, ALU operations between registers in the Register File are executed. The ALU operations are divided into three main categories – arithmetic, logic and bit-functions. Some microcontrollers in the AVR product family feature a hardware multiplier in the arithmetic part of the ALU.

The Flash Program Memory

The ATtiny15L contains 1K byte On-chip, In-System Programmable Flash memory for program storage. Since all instructions are single 16-bit words, the Flash is organized as 512 x 16 words. The Flash memory has an endurance of at least 1,000 write/erase cycles.

The ATtiny15L Program Counter is nine bits wide, thus addressing the 512 words Flash Program memory.

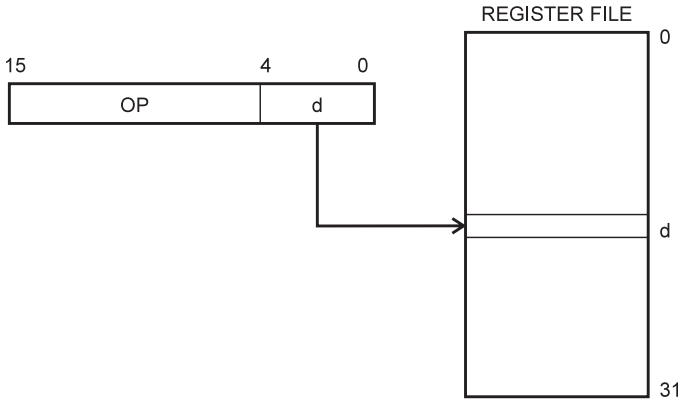
See page 54 for a detailed description on Flash memory programming.

The Program and Data Addressing Modes

The ATtiny15L AVR RISC Microcontroller supports powerful and efficient addressing modes. This section describes the various addressing modes supported in the ATtiny15L. 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

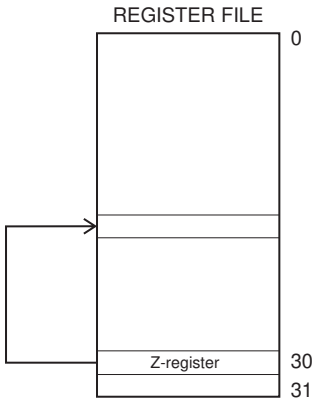
Figure 4. Direct Single-register Addressing



The operand is contained in register d (Rd).

Register Indirect

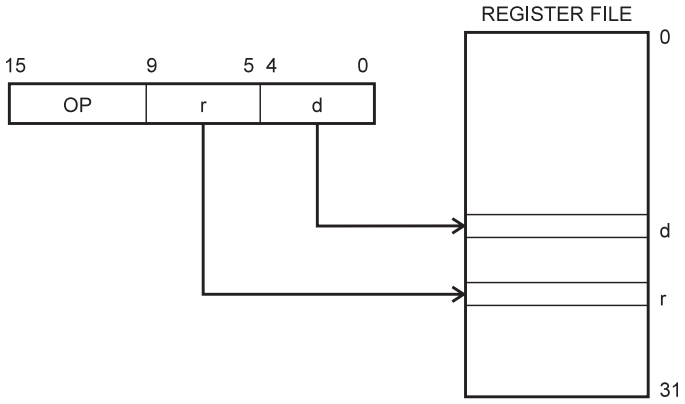
Figure 5. Indirect Register Addressing



The register accessed is the one pointed to by the Z-register low byte (R30).

Register Direct, Two Registers Rd and Rr

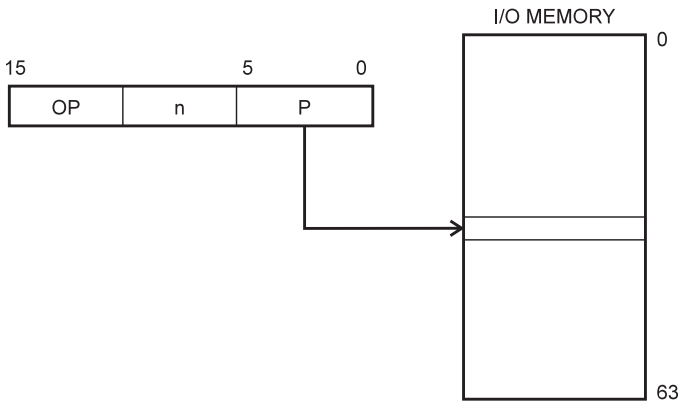
Figure 6. Direct Register Addressing, Two Registers



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

I/O Direct

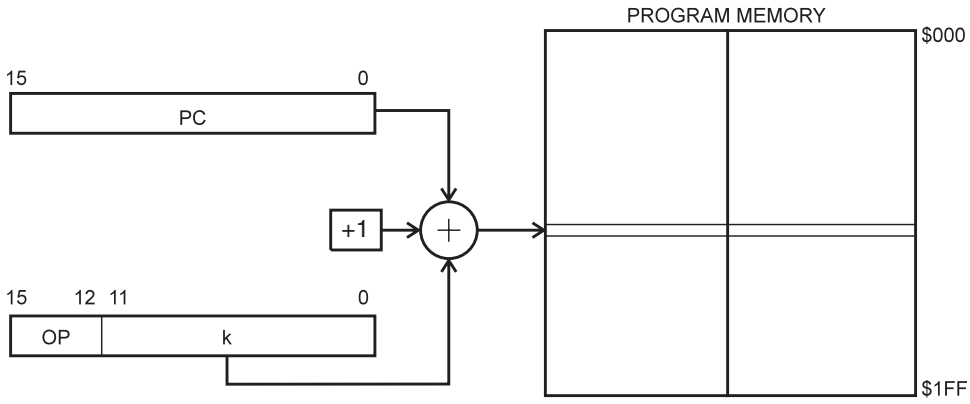
Figure 7. I/O Direct Addressing



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

Relative Program Addressing, RJMP and RCALL

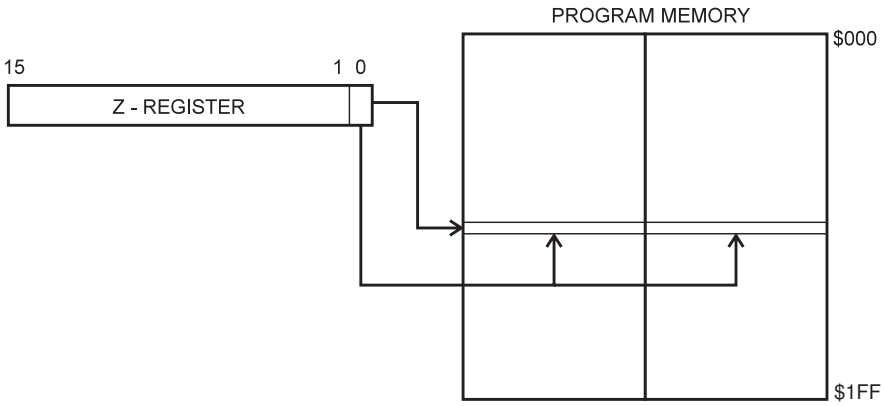
Figure 8. Relative Program Memory Addressing



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

Constant Addressing using the LPM Instruction

Figure 9. Code Memory Constant Addressing



Constant byte address is specified by the Z-register contents. The 15 MSBs select word address (0 - 511), and LSB selects low byte if cleared (LSB = 0) or high byte if set (LSB = 1).

Subroutine and Interrupt Hardware Stack

The ATtiny15L uses a 3-level-deep Hardware Stack for subroutines and interrupts. The Hardware Stack is nine bits wide and stores the Program Counter (PC) return address while subroutines and interrupts are executed.

RCALL instructions and interrupts push the PC return address onto Stack level 0, and the data in the other Stack levels 1 - 2 are pushed one level deeper in the Stack. When a RET or RETI instruction is executed the returning PC is fetched from Stack level 0, and the data in the other Stack levels 1 - 2 are popped one level in the Stack.

If more than three subsequent subroutine calls or interrupts are executed, the first values written to the Stack are overwritten. Pushing four return addresses A1, A2, A3, and A4 followed by four subroutine or interrupt returns, will pop A4, A3, A2, and once more A2 from the Hardware Stack.

The EEPROM Data Memory

The ATtiny15L contains 64 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described on page 36, specifying the EEPROM Address Register, the EEPROM Data Register, and the EEPROM Control Register.

Memory Access and Instruction Execution Timing

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

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

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

Figure 10. The Parallel Instruction Fetches and Instruction Executions

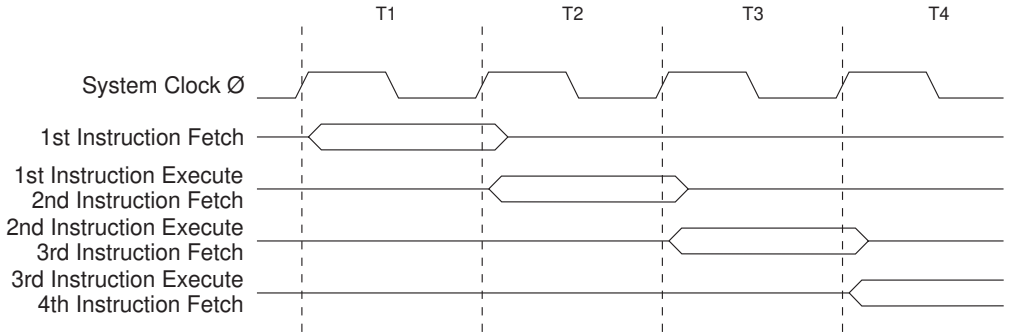
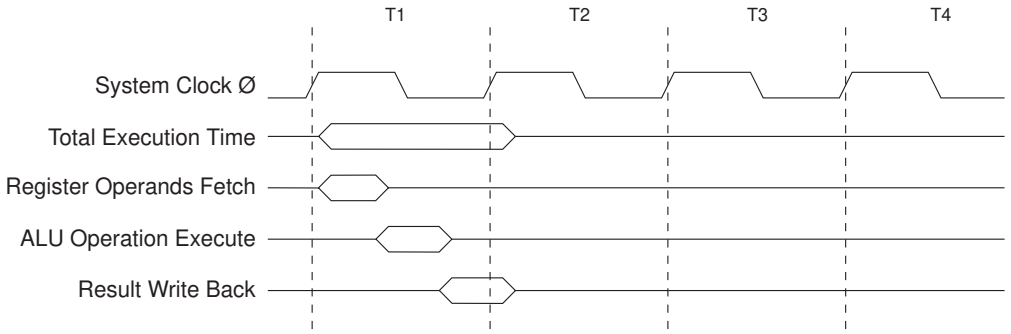


Figure 11 shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 11. Single Cycle ALU Operation



I/O Memory

The I/O space definition of the ATtiny15L is shown in Table 2.

Table 2. ATtiny15L I/O Space[1]

Address Hex	Name	Function
$3F	SREG	Status Register
$3B	GIMSK	General Interrupt Mask Register
$3A	GIFR	General Interrupt Flag Register
$39	TIMSK	Timer/Counter Interrupt Mask Register
$38	TIFR	Timer/Counter Interrupt Flag Register
$35	MCUCR	MCU Control Register
$34	MCUSR	MCU Status Register
$33	TCCR0	Timer/Counter0 Control Register
$32	TCNT0	Timer/Counter0 (8-bit)
$31	OSCCAL	Oscillator Calibration Register
$30	TCCR1	Timer/Counter1 Control Register
$2F	TCNT1	Timer/Counter1 (8-bit)
$2E	OCR1A	Timer/Counter1 Output Compare Register A
$2D	OCR1B	Timer/Counter1 Output Compare Register B
$2C	SFIOR	Special Function I/O Register
$21	WDTCR	Watchdog Timer Control Register
$1E	EEAR	EEPROM Address Register
$1D	EEDR	EEPROM Data Register
$1C	EECR	EEPROM Control Register
$18	PORTB	Data Register, Port B
$17	DDRB	Data Direction Register, Port B
$16	PINB	Input Pins, Port B
$08	ACSR	Analog Comparator Control and Status Register
$07	ADMUX	ADC Multiplexer Select Register

Table 2. ATtiny15L I/O Space[1] (Continued)

Address Hex	Name	Function
$06	ADCSR	ADC Control and Status Register
$05	ADCH	ADC Data Register High
$04	ADCL	ADC Data Register Low

Note: 1. Reserved and unused locations are not shown in the table.

All ATtiny15L I/O and peripheral registers are placed in the I/O space. The I/O locations are accessed by the IN and OUT instructions transferring data between the 32 general purpose working registers and the I/O space. 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 chapter for more details. For compatibility with future devices, reserved bits should be written zero if accessed. Reserved I/O memory addresses should never be written.

The I/O and Peripheral Control Registers are explained in the following sections.

The Status Register – SREG

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

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

- **Bit 7 – I: Global Interrupt Enable**

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

- **Bit 6 – T: Bit Copy Storage**

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

- **Bit 5 – H: Half-carry Flag**

The Half-carry Flag H indicates a half-carry in some arithmetic operations. See the Instruction Set description for detailed information.

- **Bit 4 – S: Sign Bit, S = N ⊕ V**

The S-bit is always an exclusive or between the Negative Flag N and the Two's Complement Overflow Flag V. See the Instruction Set description for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the Instruction Set description for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set description for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set description for detailed information.

Reset and Interrupt Handling

The ATtiny15L provides eight interrupt sources. These interrupts and the separate Reset Vector each have a separate Program Vector in the Program memory space. All the interrupts are assigned individual enable bits 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 3. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 (the External Interrupt Request 0), etc.

Table 3. Reset and Interrupt Vectors

Vector No.	Program Address	Source	Interrupt Definition
1	$000	RESET	External Reset, Power-on Reset, Brown-out Reset, and Watchdog Reset
2	$001	INT0	External Interrupt Request 0
3	$002	I/O Pins	Pin Change Interrupt
4	$003	TIMER1, COMPA	Timer/Counter1 Compare Match A
5	$004	TIMER1, OVF	Timer/Counter1 Overflow
6	$005	TIMER0, OVF	Timer/Counter0 Overflow
7	$006	EE_RDY	EEPROM Ready
8	$007	ANA_COMP	Analog Comparator
9	$008	ADC	ADC Conversion Complete

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

```
Address   Labels    Code                    Comments
$000              rjmp     RESET          ; Reset handler
$001              rjmp     EXT_INT0       ; IRQ0 handler
$002              rjmp     PIN_CHANGE     ; Pin change handler
$003              rjmp     TIM1_CMP       ; Timer1 compare match
$004              rjmp     TIM1_OVF       ; Timer1 overflow handler
$005              rjmp     TIM0_OVF       ; Timer0 overflow handler
$006              rjmp     EE_RDY         ; EEPROM Ready handler
$007              rjmp     ANA_COMP       ; Analog Comparator handler
$008              rjmp     ADC            ; ADC Conversion Handler
;
$009      MAIN:   <instr>  xxx            ; Main program start
  …         …        …      …
```

ATtiny15L Reset Sources

The ATtiny15L has four sources of Reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POR}).
- External Reset. The MCU is reset when a low-level is present on the $\overline{\text{RESET}}$ pin for more than 500 ns.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires, and the Watchdog is enabled.
- Brown-out Reset. The MCU is reset when the supply voltage V_{CC} is below the Brown-out Reset threshold (V_{BOT}).

During Reset, all I/O Registers are then set to their initial values, and the program starts execution from address $000. The instruction placed in address $000 must be an RJMP (relative jump) instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 12 shows the reset logic. Table 4 and Table 5 define the timing and electrical parameters of the reset circuitry. Note that the Register File is unchanged by a reset.

Figure 12. Reset Logic

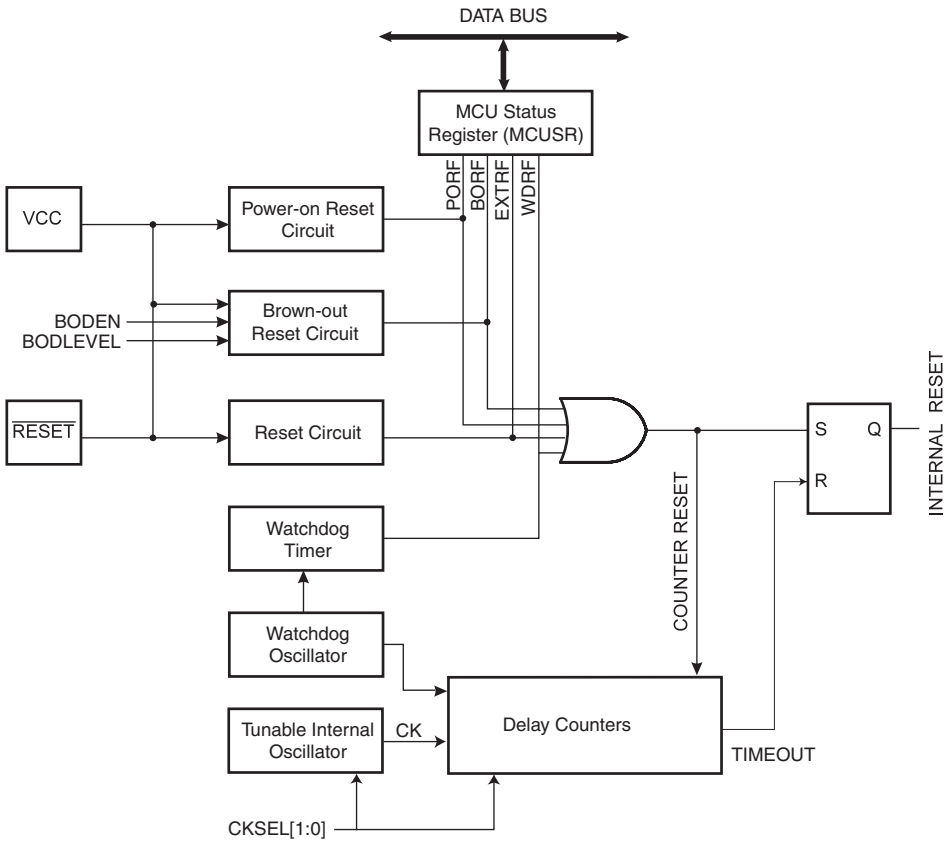


Table 4. Reset Characteristics (V_{CC} = 5.0V)[1]

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)	BOD disabled	1.0	1.4	1.8	V
		BOD enabled	1.7	2.2	2.7	V
	Power-on Reset Threshold Voltage (falling)[1]	BOD disabled	0.4	0.6	0.8	V
		BOD enabled	1.7	2.2	2.7	V
V_{RST}	$\overline{\text{RESET}}$ Pin Threshold Voltage		–	–	0.85 V_{CC}	V
V_{BOT}	Brown-out Reset Threshold Voltage	(BODLEVEL = 1)	2.3	2.7	2.9	V
		(BODLEVEL = 0)	3.4	4.0	4.3	V

Note: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling).

Table 5. Reset Delay Selections[1]

BODEN[2]	CKSEL [1:0][2]	Start-up Time, t_{TOUT} at V_{CC} = 2.7V	Start-up Time, t_{TOUT} at V_{CC} = 5.0V	Recommended Usage
x	00	256 ms + 18 CK	64 ms + 18 CK	BOD disabled, slowly rising power
x	01	256 ms + 18 CK	64 ms + 18 CK	BOD disabled, slowly rising power
x	10	16 ms + 18 CK	4 ms + 18 CK	BOD disabled, quickly rising power
1	11	18 CK + 32 µs	18 CK + 8 µs	BOD disabled
0	11	18 CK + 128 µs	18 CK + 32 µs	BOD enabled

Notes: 1. On Power-up, the start-up time is increased with typical 0.6 ms.
2. "0" means programmed, "1" means unprogrammed.

Table 5 shows the start-up times from Reset. When the CPU wakes up from Power-down, only the clock-counting part of the start-up time is used. The Watchdog Oscillator is used for timing the real-time part of the start-up time. The number Watchdog Oscillator cycles used for each time-out is shown in Table 6.

The frequency of the Watchdog Oscillator is voltage dependent as shown in the Electrical Characteristics section on page 64. The device is shipped with CKSEL = "00".

Table 6. Number of Watchdog Oscillator Cycles

V_{CC} Conditions	Time-out	Number of Cycles
2.7V	32 µs	8
2.7V	128 µs	32
2.7V	16 ms	4K
2.7V	256 ms	64K
5.0V	8 µs	8
5.0V	32 µs	32
5.0V	4 ms	4K
5.0V	64 ms	64K

Power-on Reset

A Power-on Reset (POR) pulse is generated by an On-chip Detection circuit. The detection level is nominally defined in Table 4. The POR is activated whenever V_{CC} is below the detection level. The POR circuit can be used to trigger the Start-up Reset, as well as detect a failure in supply voltage.

A Power-on Reset (POR) circuit ensures that the device is Reset from Power-on. Reaching the Power-on Reset threshold voltage invokes a delay counter, which determines the delay, for which the device is kept in RESET after V_{CC} rise. The Time-out period of the delay counter can be defined by the user through the CKSEL Fuses. The different selections for the delay period are presented in Table 5. The RESET signal is activated again, without any delay, when the V_{CC} decreases below detection level.

Figure 13. "MCU Start-up, $\overline{\text{RESET}}$ Tied to V_{CC}

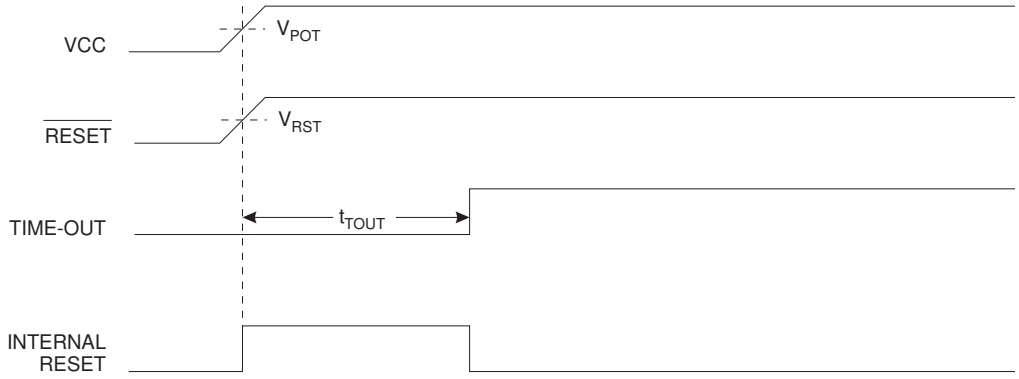
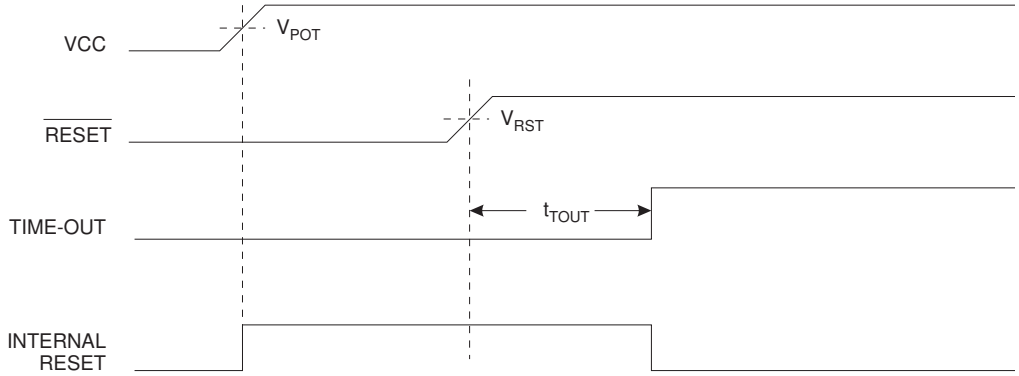


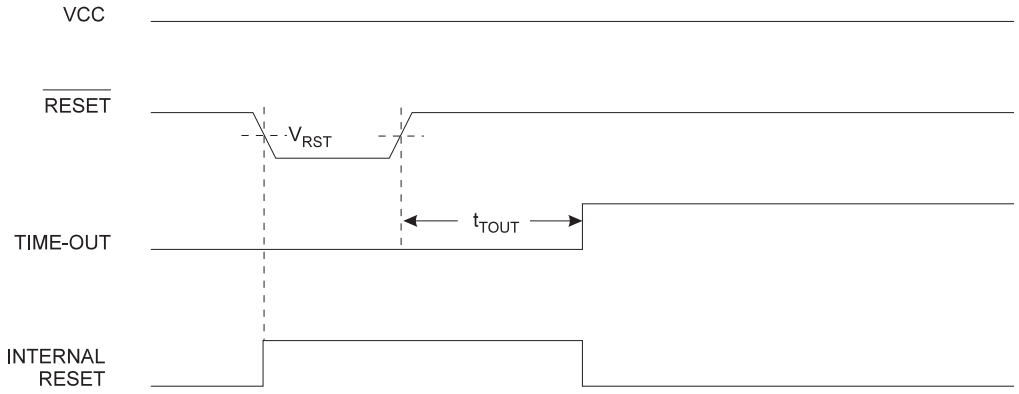
Figure 14. MCU Start-up, $\overline{\text{RESET}}$ Extended Externally



External Reset

An External Reset is generated by a low-level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than 500 ns will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage (V_{RST}) on its positive edge, the delay timer starts the MCU after the Time-out period t_{TOUT} has expired.

Figure 15. External Reset during Operation

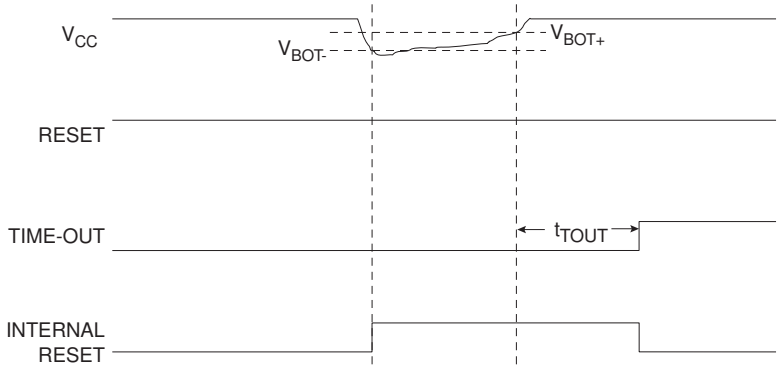


Brown-out Detection

ATtiny15L has an On-chip Brown-out Detection (BOD) circuit for monitoring the V_{CC} level during the operation. The BOD circuit can be enabled/disabled by the fuse BODEN. When BODEN is enabled (BODEN programmed), and V_{CC} decreases below the trigger level, the Brown-out Reset is immediately activated. When V_{CC} increases above the trigger level, the Brown-out Reset is deactivated after a delay. The delay is defined by the user in the same way as the delay of POR signal, in Table 5. The trigger level for the BOD can be selected by the fuse BODLEVEL to be 2.7V (BODLEVEL unprogrammed), or 4.0V (BODLEVEL programmed). The trigger level has a hysteresis of 50 mV to ensure spike-free Brown-out Detection.

The BOD circuit will only detect a drop in V_{CC} if the voltage stays below the trigger level for longer than 3 µs for trigger level 4.0V, 7 µs for trigger level 2.7V (typical values).

Figure 16. Brown-out Reset during Operation(1)

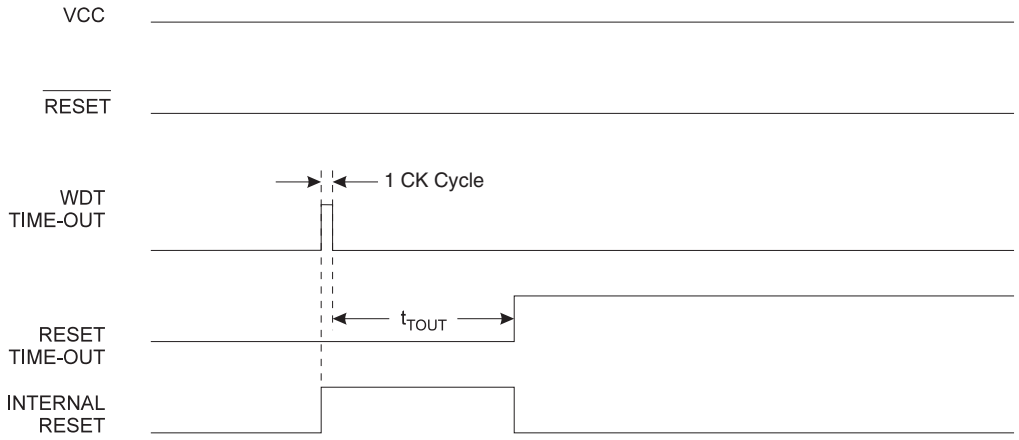


Note: 1. The hysteresis on V_{BOT}: $V_{BOT+} = V_{BOT} + 25$ mV, $V_{BOT-} = V_{BOT} - 25$ mV.

Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the Time-out period t_{TOUT}. Refer to page 34 for details on operation of the Watchdog Timer.

Figure 17. Watchdog Reset during Operation



MCU Status Register – MCUSR

The MCU Status Register provides information on which reset source caused an MCU Reset.

Bit	7	6	5	4	3	2	1	0	
$34	–	–	–	–	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	See Bit Description				

- **Bit 7..4 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny15L and always read as zero.

- **Bit 3 – WDRF: Watchdog Reset Flag**

This bit is set (one) if a Watchdog Reset occurs. The bit is reset (zero) by a Power-on Reset, or by writing a logical “0” to the flag.

- **Bit 2 – BORF: Brown-out Reset Flag**

This bit is set (one) if a Brown-out Reset occurs. The bit is reset (zero) by a Power-on Reset, or by writing a logical “0” to the flag.

- **Bit 1 – EXTRF: External Reset Flag**

This bit is set (one) if a External Reset occurs. The bit is reset (zero) by a Power-on Reset, or by writing a logical “0” to the flag.

- **Bit 0 – PORF: Power-on Reset Flag**

This bit is set (one) if a Power-on Reset occurs. The bit is reset (zero) by writing a logical “0” to the flag.

To make use of the Reset Flags to identify a reset condition, the user should read and then reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags.

Internal Voltage Reference

ATtiny15L features an internal bandgap reference with a nominal voltage of 1.22V. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator. The 2.56V reference to the ADC is generated from the internal bandgap reference.

Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The maximum start-up time is 10 µs. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODEN Fuse).
2. When the bandgap reference is connected to the Analog Comparator (by setting the AINBG bit in ACSR).
3. When the ADC is enabled.

Thus, when the BOD is not enabled, after setting the AINBG bit, the user must always allow the reference to start-up before the output from the Analog Comparator is used. The bandgap reference uses typically 10 µA, and to reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

Interrupt Handling

The ATtiny15L has two 8-bit Interrupt Mask Control Registers: GIMSK (General Interrupt Mask Register) and TIMSK (Timer/Counter Interrupt Mask Register).

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared (zero) and all interrupts are disabled. The user software can set the I-bit (one) to enable interrupts. The I-bit is set (one) when a Return from Interrupt instruction (RETI) is executed.

When the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, hardware clears the corresponding flag that generated the interrupt. Some of the interrupt flags can also be cleared by writing a logical “1” to the flag bit position(s) to be cleared.

If an interrupt condition occurs when the corresponding interrupt enable bit is cleared (zero), the interrupt flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software.

If one or more interrupt conditions occur when the global interrupt enable bit is cleared (zero), the corresponding interrupt flag(s) will be set and remembered until the global interrupt enable bit is set (one), and will be executed by order of priority.

Note that external level interrupt does not have a flag, and will only be remembered for as long as the interrupt condition is present.

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

Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After the four clock cycles the Program Vector address for the actual interrupt handling routine is executed. During this 4-clock-cycle period, the Program Counter (nine bits) is pushed onto the Stack. The vector is often a relative jump to the interrupt routine, and this jump takes two clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the Program Counter (nine bits) is popped back from the Stack. When AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

The General Interrupt Mask Register – GIMSK

Bit	7	6	5	4	3	2	1	0	
$3B	–	INT0	PCIE	–	–	–	–	–	GIMSK
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bit 6 – INT0: External Interrupt Request 0 Enable**

When the INT0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the external pin interrupt is activated. The Interrupt Sense Control0 bits 1/0 (ISC01 and ISC00) in the MCU general Control Register (MCUCR) define whether the external interrupt is activated on rising or falling edge, on pin change, or low level of the INT0 pin. Activity on the pin will cause an interrupt request even if INT0 is configured as an output.

The corresponding interrupt of External Interrupt Request 0 is executed from Program memory address $001. See also “External Interrupts.”

- **Bit 5 – PCIE: Pin Change Interrupt Enable**

When the PCIE bit is set (one) and the I-bit in the Status Register (SREG) is set (one), the interrupt on pin change is enabled. Any change on any input or I/O pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from Program memory address $002. See also “Pin Change Interrupt.”

- **Bits 4..0 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny15L and always read as zero.

The General Interrupt Flag Register – GIFR

Bit	7	6	5	4	3	2	1	0	
$3A	–	INTF0	PCIF	–	–	–	–	–	GIFR
Read/Write	R	R/W	R/W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bit 6 – INTF0: External Interrupt Flag0**

When an edge or logic change on the INT0 pin triggers an interrupt request, INTF0 becomes set (one). If the I-bit in SREG and the INT0 bit in GIMSK are set (one), the MCU will jump to the Interrupt Vector at address $001. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical “1” to it. The flag is always cleared when INT0 is configured as level interrupt.

- **Bit 5 – PCIF: Pin Change Interrupt Flag**

When an event on any input or I/O pin triggers an interrupt request, PCIF becomes set (one). If the I-bit in SREG and the PCIE bit in GIMSK are set (one), the MCU will jump to the Interrupt Vector at address $002. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical “1” to it.

- **Bits 4..0 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny15L and always read as zero.

The Timer/Counter Interrupt Mask Register – TIMSK

Bit	7	6	5	4	3	2	1	0	
$39	–	OCIE1A	–	–	–	TOIE1	TOIE0	–	TIMSK
Read/Write	R	R/W	R	R	R	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bit 6 – OCIE1A: Timer/Counter1 Output Compare Interrupt Enable**

When the OCIE1A bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Compare Match, interrupt is enabled. The corresponding interrupt (at

vector $003) is executed if a compare match A in Timer/Counter1 occurs, i.e., when the OCF1A bit is set (one) in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 5..3 – Res: Reserved Bits**

These bits are reserved bits in the ATtiny15L and always read as zero.

- **Bit 2 – TOIE1: Timer/Counter1 Overflow Interrupt Enable**

When the TOIE1 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter1 Overflow interrupt is enabled. The corresponding interrupt (at vector $004) is executed if an overflow in Timer/Counter1 occurs, i.e., when the TOV1 bit is set (one) in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 1 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is set (one) and the I-bit in the Status Register is set (one), the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt (at vector $005) is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set (one) in the Timer/Counter Interrupt Flag Register (TIFR).

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

The Timer/Counter Interrupt Flag Register – TIFR

Bit	7	6	5	4	3	2	1	0	
$38	–	OCF1A	–	–	–	TOV1	TOV0	–	TIFR
Read/Write	R	R/W	R	R	R	R/W	R/W	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bit 6 – OCF1A: Output Compare Flag 1A**

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

- **Bits 5..3 – Res: Reserved bits**

These bits are reserved bits in the ATtiny15L and always read as zero.

- **Bit 2 – TOV1: Timer/Counter1 Overflow Flag**

The bit TOV1 is set (one) when an overflow occurs in Timer/Counter1. TOV1 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV1 is cleared by writing a logical “1” to the flag. When the SREG I-bit, TOIE1 (Timer/Counter1 Overflow Interrupt Enable) and TOV1 are set (one), the Timer/Counter1 Overflow Interrupt is executed.

- **Bit 1 – TOV0: Timer/Counter0 Overflow Flag**

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

- **Bit 0 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

External Interrupt

The External Interrupt is triggered by the INT0 pin. Observe that, if enabled, the interrupt will trigger even if the INT0 pin is configured as an output. This feature provides a way of generating a software interrupt. The External Interrupt can be triggered by a falling or rising edge, a pin change, or a low level. This is set up as indicated in the specification for the MCU Control Register (MCUCR). When the external interrupt is enabled and is configured as level-triggered, the interrupt will trigger as long as the pin is held low.

The External Interrupt is set up as described in the specification for the MCU Control Register (MCUCR).

Pin Change Interrupt

The pin change interrupt is triggered by any change in logical value on any input or I/O pin. Change on pins PB4..0 will always cause an interrupt. Change on pin PB5 will cause an interrupt if the pin is configured as input or I/O, as described in the section “Pin Descriptions” on page 4. Observe that, if enabled, the interrupt will trigger even if the changing pin is configured as an output. This feature provides a way of generating a software interrupt. Also observe that the pin change interrupt will trigger even if the pin activity triggers another interrupt, for example the external interrupt. This implies that one external event might cause several interrupts. The values on the pins are sampled before detecting edges. If pin change interrupt is enabled, pulses that last longer than one CPU clock period will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt.

The MCU Control Register – MCUCR

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

Bit	7	6	5	4	3	2	1	0	
$35	–	PUD	SE	SM1	SM0	–	ISC01	ISC00	MCUCR
Read/Write	R	R/W	R/W	R/W	R/W	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bit 6- PUD: Pull-up Disable**

This PUD bit must be set (one) to disable internal pull-up registers at Port B.

- **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**

These bits select between the three available sleep modes, as shown in Table 7.

Table 7. Sleep Modes

SM1	SM0	Sleep Mode
0	0	Idle mode
0	1	ADC Noise Reduction mode
1	0	Power-down mode
1	1	Reserved

For details, refer to “Sleep Modes” below.

- **Bit 2 – Res: Reserved Bit**

This bit is a reserved bit in the ATtiny15L and always reads as zero.

- **Bits 1, 0 – ISC01, ISC00: Interrupt Sense Control 0 Bit 1 and Bit 0**

The External Interrupt 0 is activated by the external pin INT0 if the SREG I-flag and the corresponding interrupt mask is set (one). The activity on the external INT0 pin that activates the interrupt is defined in Table 8:

Table 8. Interrupt 0 Sense Control(1)

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

Note: 1. When changing the ISC10/ISC00 bits, INT0 must be disabled by clearing its Interrupt Enable bit in the GIMSK Register. Otherwise an interrupt can occur when the bits are changed.

Sleep Modes

To enter any of the three sleep modes, the SE bit in MCUCR must be set (one) and a SLEEP instruction must be executed. The SM1 and SM0 bits in the MCUCR Register select which sleep mode (Idle, ADC Noise Reduction or Power-down) will be activated by the SLEEP instruction (see Table 7). If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles, executes the interrupt routine and resumes execution from the instruction following SLEEP. On wake-up from Power-down mode on pin change, two instruction cycles are executed before the Pin Change Interrupt Flag is updated. The contents of the Register File, SRAM, and I/O memory are unaltered when the device wakes up from sleep. If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

Idle Mode

When the SM1/SM0 bits are “00”, the SLEEP instruction forces the MCU into the Idle mode, stopping the CPU but allowing the ADC, Analog Comparator, Timer/Counters, Watchdog and the Interrupt system to continue operating. This enables the MCU to wake-up from external triggered interrupts as well as internal ones like the Timer Overflow Interrupt and Watchdog Reset. If the ADC is enabled, a conversion starts automatically when this mode is entered. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ADC-bit in the Analog Comparator Control and Status Register (ACSR). This will reduce power consumption in Idle mode.

ADC Noise Reduction Mode

When the SM1/SM0 bits are “01”, the SLEEP instruction forces the MCU into the ADC Noise Reduction mode, stopping the CPU but allowing the ADC, the external interrupt pin, pin change interrupt and the Watchdog (if enabled) to continue operating. Please note that the clock system including the PLL is also active in the ADC Noise Reduction mode. This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. In addition to Watchdog Time-out and External Reset, only an external level-triggered interrupt, a pin change interrupt or an ADC interrupt can wake up the MCU.

Power-down Mode

When the SM1/SM0 bits are “10”, the SLEEP instruction forces the MCU into the Power-down mode. Only an External Reset, a Watchdog Reset (if enabled), an external level-triggered interrupt, or a pin change interrupt can wake up the MCU.

Note that if a level-triggered or pin change interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. This makes the MCU less sensitive to noise. The changed level is sampled twice by the Watchdog Oscillator clock, and if the input has the required level during this time, the MCU will wake up. The period of the Watchdog Oscillator is 2.9 µs (nominal) at 3.0V and 25°C. The frequency of the Watchdog Oscillator is voltage-dependent as shown in the “Electrical Characteristics” section.

When waking up from the Power-down mode, a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after having been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out period.

Tuneable Internal RC Oscillator

The internal RC Oscillator provides a fixed 1.6 MHz clock (nominal at 5V and 25°C). This internal clock is always the system clock of the ATtiny15L. This Oscillator can be calibrated by writing the calibration byte (see page 55) to the OSCCAL Register.

The System Clock Oscillator Calibration Register – OSCCAL

Bit	7	6	5	4	3	2	1	0	
$31	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
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	

Writing the calibration byte to this address will trim the internal Oscillator frequency in order to remove process variations. When OSCCAL is zero (initial value), the lowest available frequency is chosen. Writing non-zero values to this register will increase the frequency of the internal oscillator. Writing $FF to the register selects the highest available frequency.

Internal PLL for Fast Peripheral Clock Generation

The internal PLL in ATtiny15L generates a clock frequency that is 16x multiplied from the RC Oscillator system clock. If the RC Oscillator frequency is the nominal 1.6 MHz, the fast peripheral clock is 25.6 MHz. The fast peripheral clock, or a clock prescaled from that, can be selected as the clock source for Timer/Counter1.

The PLL is locked on the tunable internal RC Oscillator and adjusting the tunable internal RC oscillator via the OSCCAL Register will adjust the fast peripheral clock at the same time. Timer1 may malfunction if the internal RC oscillator is adjusted beyond 1.75 MHz.

It is recommended not to take the OSCCAL adjustments to a higher frequency than 1.75 MHz in order to keep proper operation of all chip functions.

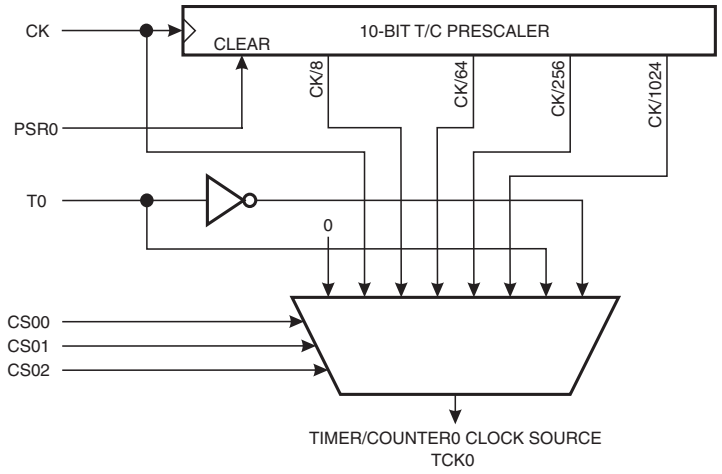
Timer/Counters

The ATtiny15L provides two general purpose 8-bit Timer/Counters. The Timer/Counters have separate prescaling selection from separate 10-bit prescalers. The Timer/Counter0 uses internal clock (CK) as the clock time base. The Timer/Counter1 may use either the internal clock (CK) or the fast peripheral clock (PCK) as the clock time base.

The Timer/Counter0 Prescaler

Figure 18 shows the Timer/Counter prescaler.

Figure 18. Timer/Counter0 Prescaler



The four prescaled selections are: CK/8, CK/64, CK/256, and CK/1024, where CK is the Oscillator clock. CK, external source and stop, can also be selected as clock sources. Setting the PSR10 bit in SFIOR resets the prescaler. This allows the user to operate with a predictable prescaler.

The Timer/Counter1 Prescaler

Figure 19 shows the Timer/Counter1 prescaler. For Timer/Counter1 the clock selections are: PCK, PCK/2, PCK/4, PCK/8, CK (=PCK/16), CK/2, CK/4, CK/8,CK/16, CK/32, CK/64, CK/128, CK/256, CK/512, CK/1024, and stop. The clock options are described in Table 12 on page 31 and the Timer/Counter1 Control Register (TCCR1). Setting the PSR1 bit in the SFIOR Register resets the 10-bit prescaler. This allows the user to operate with a predictable prescaler.

Figure 19. Timer/Counter1 Prescaler