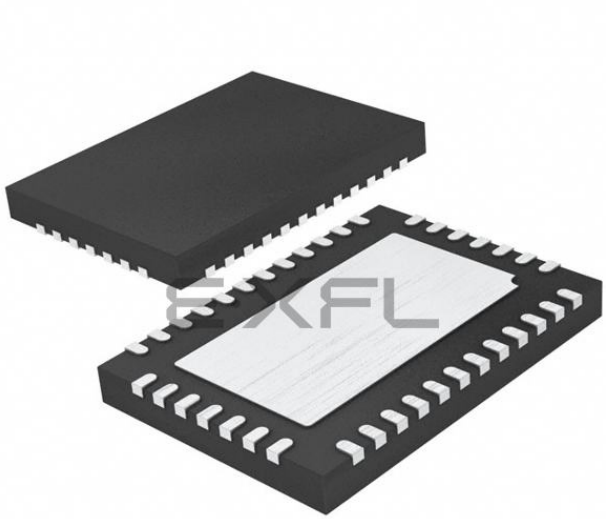


Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	I²C, LINbus, SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	16
Program Memory Size	8KB (8K x 8)
Program Memory Type	FLASH
EEPROM Size	512 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 125°C (TC)
Mounting Type	Surface Mount
Package / Case	38-VFQFN Exposed Pad
Supplier Device Package	38-VQFN (5x7)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/ata6616c-p3qw

Assembly Code Example

```
in     r16, SREG     ; store SREG value
cli    ; disable interrupts during timed sequence
sbi    EECR, EEMPE   ; start EEPROM write
sbi    EECR, EEPE
out    SREG, r16     ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG;/* store SREG value */
/* disable interrupts during timed sequence */
_CLI();
EECR |= (1<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

When using the SEI instruction to enable interrupts, the instruction following SEI will be executed before any pending interrupts, as shown in this example.

Assembly Code Example

```
sei    ; set Global Interrupt Enable
sleep  ; enter sleep, waiting for interrupt
; note: will enter sleep before any pending
; interrupt(s)
```

C Code Example

```
_SEI();      /* set Global Interrupt Enable */
_SLEEP();    /* enter sleep, waiting for interrupt */
/* note: will enter sleep before any pending interrupt(s) */
```

4.3.7.2 Interrupt Response Time

The interrupt execution response for all the enabled AVR® interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the program counter is pushed onto the stack. The vector is normally a jump to the interrupt routine, and this jump takes three 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. This increase comes in addition to the start-up time from the selected sleep mode.

A return from an interrupt handling routine takes four clock cycles. During these four clock cycles, the program counter (two bytes) is popped back from the stack, the stack pointer is incremented by two, and the I-bit in SREG is set.

Table 4-15. Clock Command List

Clock Command	CLKC3..0
No command	0000_b
Disable clock source	0001_b
Enable clock source	0010_b
Request for clock availability	0011_b
Clock source switch	0100_b
Recover system clock source code	0101_b
Enable watchdog in automatic reload mode	0110_b
CKOUT command	0111_b
No command	$1xxx_b$

4.5.5.4 CLKSELR - Clock Selection Register

Bit	7	6	5	4	3	2	1	0	
	-	COUT	CSUT1	CSUT0	CSEL3	CSEL2	CSEL1	CSEL0	CLKSELR
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	~ (CKOUT) fuse	SUT1..0 fuses			CKSEL3..0 fuses			

- **Bit 7– Res: Reserved Bit**

This bit is reserved bit in the Atmel® ATtiny87/167 and will always read as zero.

- **Bit 6 – COUT: Clock Out**

The COUT bit is initialized with ~(CKOUT) Fuse bit.
The COUT bit is only used in case of '*CKOUT*' command. Refer to Section 4.5.2.7 "Clock Output Buffer" on page 52 for using.
In case of '*Recover System Clock Source*' command, COUT it is not affected (no recovering of this setting).

- **Bits 5:4 – CSUT1:0: Clock Start-up Time**

CSUT bits are initialized with the values of SUT Fuse bits.
In case of '*Enable/Disable Clock Source*' command, CSUT field provides the code of the clock start-up time. Refer to subdivisions of Section 4.5.2 "Clock Sources" on page 47 for code of clock start-up times.
In case of '*Recover System Clock Source*' command, CSUT field is not affected (no recovering of SUT code).

- **Bits 3:0 – CSEL3:0: Clock Source Select**

CSEL bits are initialized with the values of CKSEL Fuse bits.
In case of '*Enable/Disable Clock Source*', '*Request for Clock Availability*' or '*Clock Source Switch*' command, CSEL field provides the code of the clock source. Refer to Table 4-5 on page 47 and subdivisions of Section 4.5.2 "Clock Sources" on page 47 for clock source codes.
In case of '*Recover System Clock Source*' command, CSEL field contains the code of the clock source used to drive the Clock Control Unit as described in Figure 4-10 on page 46.

- **Bit 0 – 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 enabled. The interrupt sense control0 bits 1/0 (ISC01 and ISC00) in the external interrupt control register A (EICRA) define whether the external interrupt is activated on rising and/or falling edge of the INT0 pin or level sensed. 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 the INT0 interrupt vector.

4.9.3.3 External Interrupt Flag Register – EIFR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	INTF1	INTF0	EIFR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 2 – Res: Reserved Bits**

These bits are unused bits in the Atmel® ATtiny87/167, and will always read as zero.

- **Bit 1 – INTF1: External Interrupt Flag 1**

When an edge or logic change on the INT1 pin triggers an interrupt request, INTF1 becomes set (one). If the I-bit in SREG and the INT1 bit in EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT1 is configured as a level interrupt.

- **Bit 0 – INTF0: External Interrupt Flag 0**

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 EIMSK are set (one), the MCU will jump to the corresponding interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. This flag is always cleared when INT0 is configured as a level interrupt.

4.9.3.4 Pin Change Interrupt Control Register – PCICR

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7, 2 – Res: Reserved Bits**

These bits are unused bits in the Atmel ATtiny87/167, and will always read as zero.

- **Bit 1 - PCIE1: Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT15..8 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI1 interrupt vector. PCINT15..8 pins are enabled individually by the PCMSK1 register.

- **Bit 0 - PCIE0: Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the status register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7..0 pin will cause an interrupt. The corresponding interrupt of pin change interrupt request is executed from the PCI0 interrupt vector. PCINT7..0 pins are enabled individually by the PCMSK0 register.

4.10.3.4 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 4-27.

Table 4-27. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	PCINT15 (pin change interrupt 15) ADC10 (ADC input channel 10) OC1BX (output compare and PWM output B-X for Timer/Counter1) $\overline{\text{RESET}}$ (reset input pin) dW (debugWIRE I/O)
PB6	PCINT14 (pin change interrupt 14) ADC9 (ADC input channel 9) OC1AX (output compare and PWM output A-X for Timer/Counter1) INT0 (external interrupt0 input)
PB5	PCINT13 (pin change interrupt 13) ADC8 (ADC input channel 8) OC1BW (output compare and PWM output B-W for Timer/Counter1) XTAL2 (chip clock oscillator pin 2) CLKO (system clock output)
PB4	PCINT12 (pin change interrupt 12) OC1AW (output compare and PWM output A-W for Timer/Counter1) XTAL1 (chip clock oscillator pin 1) CLKI (external clock input)
PB3	PCINT11 (pin change interrupt 11) OC1BV (output compare and PWM output B-V for Timer/Counter1)
PB2	PCINT10 (pin change interrupt 10) OC1AV (output compare and PWM output A-V for Timer/Counter1) USCK (three-wire mode USI *Default* clock input) SCL (two-wire mode USI *Default* clock input)
PB1	PCINT9 (pin change interrupt 9) OC1BU (output compare and PWM output B-U for Timer/Counter1) DO (three-wire mode USI *Default* data output)
PB0	PCINT8 (pin change interrupt 8) OC1AU (output compare and PWM output A-U for Timer/Counter1) DI (three-wire mode USI *Default* data input) SDA (two-wire mode USI *Default* data input / output)

- **PCINT11/OC1BV – Port B, Bit 3**

PCINT11: Pin change interrupt, source 11.

OC1BV: Output compare and PWM output B-V for Timer/Counter1. The PB3 pin has to be configured as an output (DDB3 set (one)) to serve this function. The OC1BV pin is also the output pin for the PWM mode timer function (c.f. OC1BV bit of TCCR1D register).

- **PCINT10/OC1AV/USCK/SCL – Port B, Bit 2**

PCINT10: Pin change interrupt, source 10.

OC1AV: Output compare and PWM output A-V for Timer/Counter1. The PB2 pin has to be configured as an output (DDB2 set (one)) to serve this function. The OC1AV pin is also the output pin for the PWM mode timer function (c.f. OC1AV bit of TCCR1D register).

USCK: Three-wire mode USI clock input.

SCL: Two-wire mode USI clock input.

- **PCINT9/OC1BU/DO – Port B, Bit 1**

PCINT9: Pin change interrupt, source 9.

OC1BU: Output compare and PWM output B-U for Timer/Counter1. The PB1 pin has to be configured as an output (DDB1 set (one)) to serve this function. The OC1BU pin is also the output pin for the PWM mode timer function (c.f. OC1BU bit of TCCR1D register).

DO: Three-wire mode USI data output. Three-wire mode data output overrides PORTB1 and it is driven to the port when the data direction bit DDB1 is set. PORTB1 still enables the pull-up, if the direction is input and PORTB1 is set (one).

- **PCINT8/OC1AU/DI/SDA – Port B, Bit 0**

IPCINT8: Pin change interrupt, source 8.

OC1AU: Output compare and PWM output A-U for Timer/Counter1. The PB0 pin has to be configured as an output (DDB0 set (one)) to serve this function. The OC1AU pin is also the output pin for the PWM mode timer function (c.f. OC1AU bit of TCCR1D register).

DI: Three-wire mode USI data input. USI three-wire mode does not override normal port functions, so pin must be configure as an input for DI function.

SDA: Two-wire mode serial interface (USI) data input/output.

4.11.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the force output compare (FOC0A) bit. Forcing compare match will not set the OCF0A flag or reload/clear the timer, but the OC0A pin will be updated as if a real compare match had occurred (the COM0A1:0 bits settings define whether the OC0A pin is set, cleared or toggled).

4.11.5.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 register will block any compare match that occurs in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0A to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

4.11.5.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT0 when using the output compare channel, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0A value, the compare match will be missed, resulting in incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is downcounting.

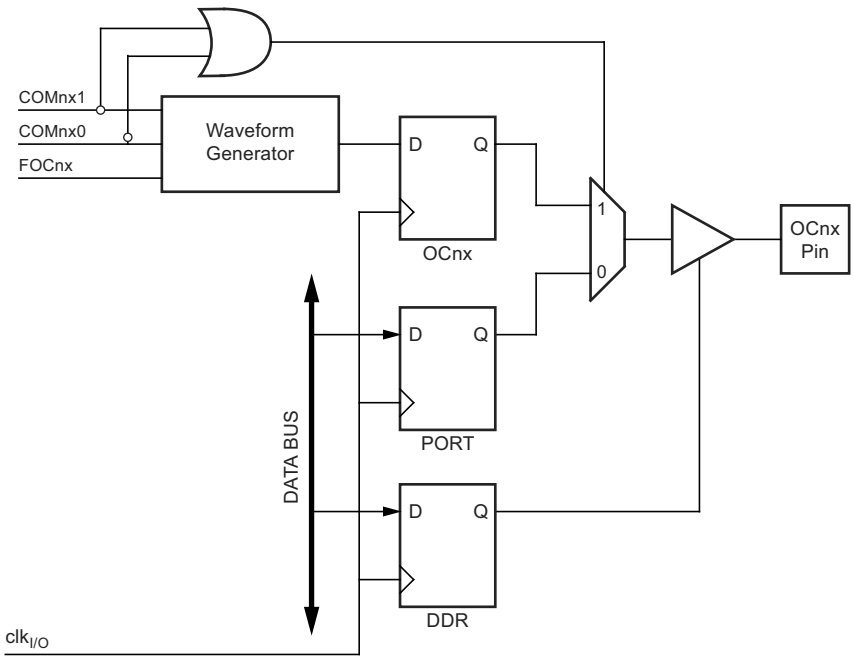
The setup of the OC0A should be performed before setting the data direction register for the port pin to output. The easiest way of setting the OC0A value is to use the force output compare (FOC0A) strobe bit in normal mode. The OC0A register keeps its value even when changing between waveform generation modes.

Be aware that the COM0A1:0 bits are not double buffered together with the compare value. Changing the COM0A1:0 bits will take effect immediately.

4.11.6 Compare Match Output Unit

The compare output mode (COM0A1:0) bits have two functions. The waveform generator uses the COM0A1:0 bits for defining the output compare (OC0A) state at the next compare match. Also, the COM0A1:0 bits control the OC0A pin output source. Figure 4-33 shows a simplified schematic of the logic affected by the COM0A1:0 bit setting. The I/O registers, I/O bits, and I/O pins in the figure are shown in bold. Only the parts of the general I/O port control registers (DDR and PORT) that are affected by the COM0A1:0 bits are shown. When referring to the OC0A state, the reference is for the internal OC0A Register, not the OC0A pin.

Figure 4-33. Compare Match Output Logic



- **Bit 0 – TCR0BUB: Timer/Counter0 Control Register B Update Busy**

When Timer/Counter0 operates asynchronously and TCCR0B is written, this bit becomes set. When TCCR0B has been updated from the temporary storage register, this bit is cleared by hardware. A logical zero in this bit indicates that TCCR0B is ready to be updated with a new value.

If a write is performed to any of the four Timer/Counter0 registers while its update busy flag is set, the updated value might get corrupted and cause an unintentional interrupt to occur.

The mechanisms for reading TCNT0, OCR0A, TCCR0A and TCCR0B are different. When reading TCNT0, the actual timer value is read. When reading OCR0A, TCCR0A or TCCR0B the value in the temporary storage register is read.

4.11.11.6 Timer/Counter0 Interrupt Mask Register – TIMSK0

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Res: Reserved Bits**

These bits are reserved in the Atmel® ATtiny87/167 and will always read as zero.

- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one and the I-bit in the status register is set (one), the Timer/Counter0 compare match A interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter0 interrupt flag register – TIFR0.

- **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one and the I-bit in the status register is set (one), the Timer/Counter0 overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs, i.e., when the TOV0 bit is set in the Timer/Counter0 interrupt flag register – TIFR0.

4.11.11.7 Timer/Counter0 Interrupt Flag Register – TIFR0

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Res: Reserved Bits**

These bits are reserved in the Atmel ATtiny87/167 and will always read as zero.

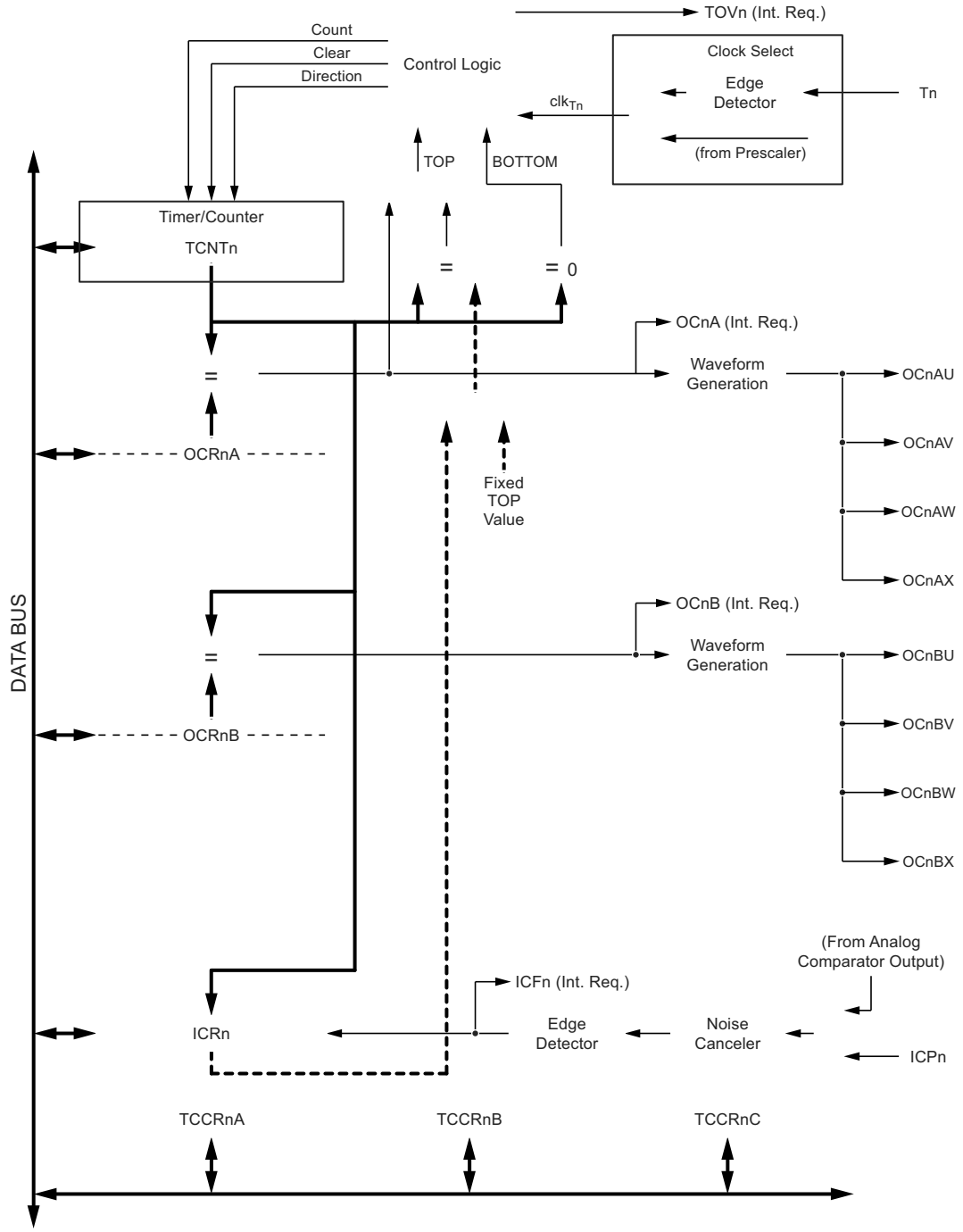
- **Bit 1 – OCF0A: Output Compare Flag 0 A**

The OCF0A bit is set (one) when a compare match occurs between the Timer/Counter0 and the data in OCR0A – output compare register0. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0 (Timer/Counter0 compare match Interrupt Enable), and OCF0A are set (one), the Timer/Counter0 compare match Interrupt is executed.

- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

The TOV0 bit 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 logic one to the flag. When the SREG I-bit, TOIE0A (Timer/Counter0 overflow interrupt enable), and TOV0 are set (one), the Timer/Counter0 Overflow interrupt is executed. In PWM mode, this bit is set when Timer/Counter0 changes counting direction at 0x00.

Figure 4-44. 16-bit Timer/Counter1 Block Diagram(1)



Note: 1. Refer to Table 4-27 on page 97, and Table 4-24 on page 92 for Timer/Counter1 pin placement and description.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the compare registers. If the TOP value is lower than any of the compare registers, a compare match will never occur between the TCNT1 and the OCR1A/B. Note that when using fixed TOP values, the unused bits are masked to zero when any of the OCR1A/B registers are written. As the third period shown in Figure 4-52 illustrates, changing the TOP actively while the Timer/Counter is running in the phase correct mode can result in an unsymmetrical output. The reason for this can be found in the time of update of the OCR1A/B register. Since the OCR1A/B update occurs at TOP, the PWM period starts and ends at TOP. This implies that the length of the falling slope is determined by the previous TOP value, while the length of the rising slope is determined by the new TOP value. When these two values differ the two slopes of the period will differ in length. The difference in length gives the unsymmetrical result on the output.

It is recommended to use the phase and frequency correct mode instead of the phase correct mode when changing the TOP value while the Timer/Counter is running. When using a static TOP value there are practically no differences between the two modes of operation.

In phase correct PWM mode, the compare units allow generation of PWM waveforms on the OC1A/B pins. Setting the COM1A/B1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1A/B1:0 to three. The actual OC1A/B value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1A/B) and OC1A/Bi is set. The PWM waveform is generated by setting (or clearing) the OC1A/B register at the compare match between OCR1A/B and TCNT1 when the counter increments, and clearing (or setting) the OC1A/B register at compare match between OCR1A/B and TCNT1 when the counter decrements. The PWM frequency for the output when using phase correct PWM can be calculated by the following equation:

$$f_{OCnxPCPWM} = \frac{f_{clk\_I/O}}{2 \times N \times TOP}$$

The N variable represents the prescaler divider (1, 8, 64, 256, or 1024).

The extreme values for the OCR1A/B register represent special cases when generating a PWM waveform output in the phase correct PWM mode. If the OCR1A/B is set equal to BOTTOM the output will be continuously low and if set equal to TOP the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

4.13.9.5 Phase and Frequency Correct PWM Mode

The phase and frequency correct pulse width modulation, or phase and frequency correct PWM mode (WGM13:0 = 8 or 9) provides a high resolution phase and frequency correct PWM waveform generation option. The phase and frequency correct PWM mode is, like the phase correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting compare output mode, the output compare (OC1A/B) is cleared on the compare match between TCNT1 and OCR1A/B while upcounting, and set on the compare match while downcounting. In inverting compare output mode, the operation is inverted. The dual-slope operation gives a lower maximum operation frequency compared to the single-slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

The main difference between the phase correct, and the phase and frequency correct PWM mode is the time the OCR1A/B register is updated by the OCR1A/B buffer register, (see Figure 4-52 on page 136 and Figure 4-53 on page 138).

The PWM resolution for the phase and frequency correct PWM mode can be defined by either ICR1 or OCR1A. The minimum resolution allowed is 2-bit (ICR1 or OCR1A set to 0x0003), and the maximum resolution is 16-bit (ICR1 or OCR1A set to MAX). The PWM resolution in bits can be calculated using the following equation:

$$R_{PFCPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In phase and frequency correct PWM mode the counter is incremented until the counter value matches either the value in ICR1 (WGM13:0 = 8), or the value in OCR1A (WGM13:0 = 9). The counter has then reached the TOP and changes the count direction. The TCNT1 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct and frequency correct PWM mode is shown on Figure 4-53. The figure shows phase and frequency correct PWM mode when OCR1A or ICR1 is used to define TOP. The TCNT1 value is in the timing diagram shown as a histogram for illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT1 slopes represent compare matches between OCR1A/B and TCNT1. The OC1A/B interrupt flag will be set when a compare match occurs.

4.13.11.6 Output Compare Register A – OCR1AH and OCR1AL

Bit	7	6	5	4	3	2	1	0	
	OCR1A[15:8]								OCR1AH
	OCR1A[7:0]								OCR1AL
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	

4.13.11.7 Output Compare Register B – OCR1BH and OCR1BL

Bit	7	6	5	4	3	2	1	0	
	OCR1B[15:8]								OCR1BH
	OCR1B[7:0]								OCR1BL
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	

The output compare registers contain a 16-bit value that is continuously compared with the counter value (TCNT1). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC1A/B pin.

The output compare registers are 16-bit in size. To ensure that both the high and low bytes are written simultaneously when the CPU writes to these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section 4.13.3 “Accessing 16-bit Registers” on page 123.

4.13.11.8 Input Capture Register – ICR1H and ICR1L

Bit	7	6	5	4	3	2	1	0	
	ICR1[15:8]								ICR1H
	ICR1[7:0]								ICR1L
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	

The input capture is updated with the counter (TCNT1) value each time an event occurs on the ICP1 pin (or optionally on the analog comparator output for Timer/Counter1). The input capture can be used for defining the counter TOP value.

The input capture register is 16-bit in size. To ensure that both the high and low bytes are read simultaneously when the CPU accesses these registers, the access is performed using an 8-bit temporary high byte register (TEMP). This temporary register is shared by all the other 16-bit registers. See Section 4.13.3 “Accessing 16-bit Registers” on page 123.

4.13.11.9 Timer/Counter1 Interrupt Mask Register – TIMSK1

Bit	7	6	5	4	3	2	1	0	
	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	TIMSK1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..6 – Reserved Bits**

These bits are reserved for future use.

- **Bit 5 – ICIE1: Input Capture Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 input capture interrupt is enabled. The corresponding interrupt vector (see Section 4.8.1 “Innterrupt Vectors in Atmel ATtiny87/167” on page 76) is executed when the ICF1 flag, located in TIFR1, is set.

- **Bit 4..3 – Reserved Bits**

These bits are reserved for future use.

- **Bit 2 – OCIE1B: Output Compare B Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare B Match interrupt is enabled. The corresponding interrupt vector (see Section 4.8.1 “Innterrupt Vectors in Atmel ATtiny87/167” on page 76) is executed when the OCF1B flag, located in TIFR1, is set.

- **Bit 1 – OCIE1A: Output Compare A Match Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 output compare A match interrupt is enabled. The corresponding interrupt vector (see Section 4.8.1 “Innterrupt Vectors in Atmel ATtiny87/167” on page 76) is executed when the OCF1A flag, located in TIFR1, is set.

- **Bit 0 – TOIE1: Timer/Counter Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the status register is set (interrupts globally enabled), the Timer/Counter1 overflow interrupt is enabled. The corresponding interrupt vector (see Section 4.8.1 “Innterrupt Vectors in Atmel ATtiny87/167” on page 76) is executed when the TOV1 flag, located in TIFR1, is set.

4.13.11.10Timer/Counter1 Interrupt Flag Register – TIFR1

Bit	7	6	5	4	3	2	1	0	
	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	TIFR1
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7..6 – Reserved Bits**

These bits are reserved for future use.

- **Bit 5 – ICF1: Input Capture Flag**

This flag is set when a capture event occurs on the ICP1 pin. When the input capture register (ICR1) is set by the WGM13:0 to be used as the TOP value, the ICF1 flag is set when the counter reaches the TOP value.

ICF1 is automatically cleared when the input capture interrupt vector is executed. Alternatively, ICF1 can be cleared by writing a logic one to its bit location.

- **Bit 4..3 – Reserved Bits**

These bits are reserved for future use.

- **Bit 2 – OCF1B: Output Compare B Match Flag**

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register B (OCR1B).

Note that a forced output compare (FOC1B) strobe will not set the OCF1B flag.

OCF1B is automatically cleared when the output compare match B interrupt vector is executed. Alternatively, OCF1B can be cleared by writing a logic one to its bit location.

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

This flag is set in the timer clock cycle after the counter (TCNT1) value matches the output compare register A (OCR1A).

Note that a forced output compare (FOC1A) strobe will not set the OCF1A flag.

OCF1A is automatically cleared when the output compare match A interrupt vector is executed. Alternatively, OCF1A can be cleared by writing a logic one to its bit location.

- **Bit 0 – TOV1: Timer/Counter Overflow Flag**

The setting of this flag is dependent of the WGM13:0 bits setting. In normal and CTC modes, the TOV1 flag is set when the timer overflows. Refer to Table 4-38 on page 142 for the TOV1 flag behavior when using another WGM13:0 bit setting.

TOV1 is automatically cleared when the Timer/Counter1 overflow interrupt vector is executed. Alternatively, TOV1 can be cleared by writing a logic one to its bit location.

4.16.6.7 LIN Data Length Register - LINDLR

Bit	7	6	5	4	3	2	1	0	
	LTXDL3	LTXDL2	LTXDL1	LTXDL0	LRXDL3	LRXDL2	LRXDL1	LRXDL0	LINDLR
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	

- **Bits 7:4 - LTXDL[3:0]: LIN Transmit Data Length**
  In LIN mode, this field gives the number of bytes to be transmitted (clamped to 8 Max).
  In UART mode this field is unused.

- **Bits 3:0 - LRXDL[3:0]: LIN Receive Data Length**
  In LIN mode, this field gives the number of bytes to be received (clamped to 8 Max).
  In UART mode this field is unused.

4.16.6.8 LIN Identifier Register - LINIDR

Bit	7	6	5	4	3	2	1	0	
	LP1	LP0	LID5 / LDL1	LID4 / LDL0	LID3	LID2	LID1	LID0	LINIDR
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:6 - LP[1:0]: Parity**
  In LIN mode:

```
LP0 = LID4 ^ LID2 ^ LID1 ^ LID0
LP1 = ! ( LID1 ^ LID3 ^ LID4 ^ LID5 )
```

  In UART mode this field is unused.

- **Bits 5:4 - LDL[1:0]: LIN 1.3 Data Length**
  In LIN 1.3 mode:
  - 00 = 2-byte response,
  - 01 = 2-byte response,
  - 10 = 4-byte response,
  - 11 = 8-byte response.

  In UART mode this field is unused.

- **Bits 3:0 - LID[3:0]: LIN 1.3 Identifier**
  In LIN 1.3 mode: 4-bit identifier.
  In UART mode this field is unused.

- **Bits 5:0 - LID[5:0]: LIN 2.1 Identifier**
  In LIN 2.1 mode: 6-bit identifier (no length transported).
  In UART mode this field is unused.

Figure 4-82. Analog to Digital Converter Block Schematic

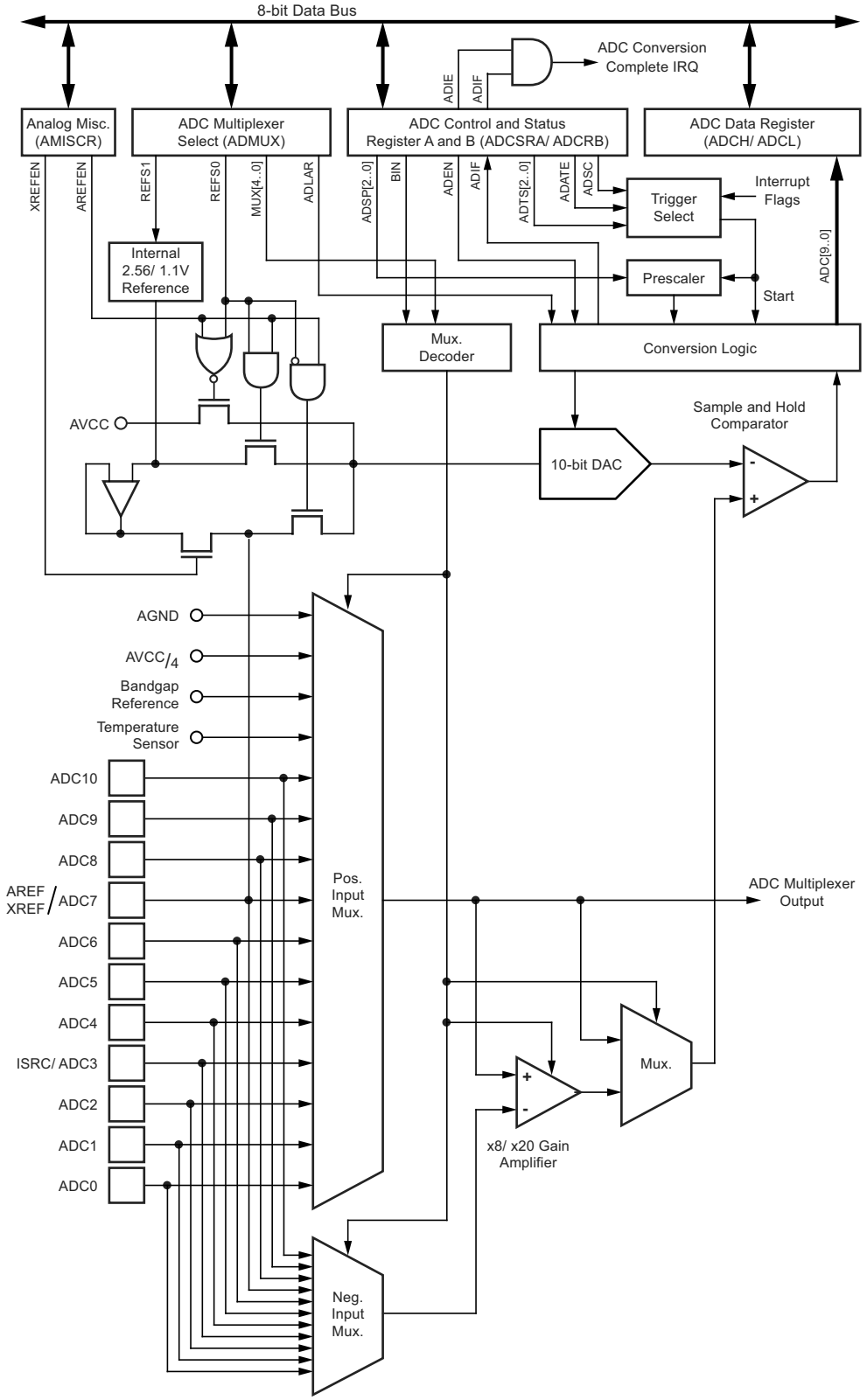
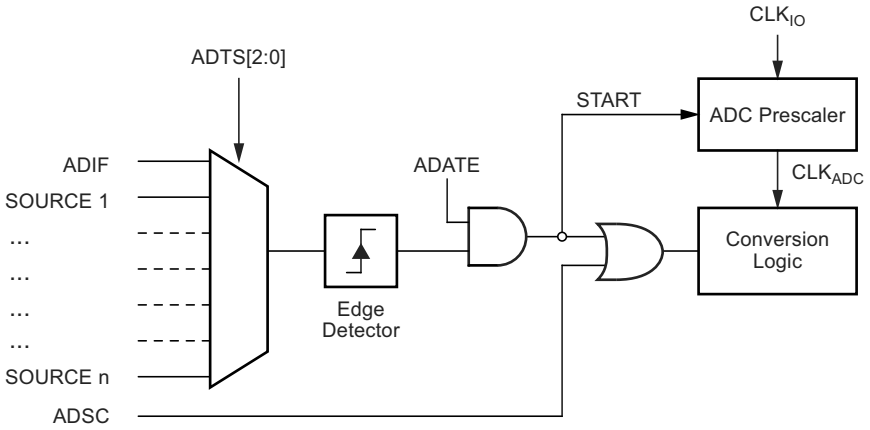


Figure 4-83. ADC Auto Trigger Logic

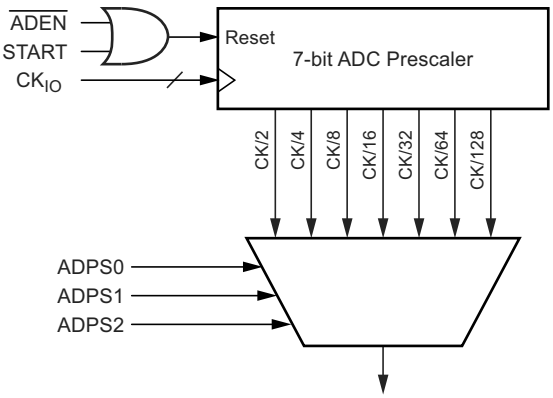


Using the ADC interrupt flag as a trigger source makes the ADC start a new conversion as soon as the ongoing conversion has finished. The ADC then operates in free running mode, constantly sampling and updating the ADC data register. The first conversion must be started by writing a logical one to the ADSC bit in ADCSRA register. In this mode the ADC will perform successive conversions independently of whether the ADC interrupt flag, ADIF is cleared or not.

If auto triggering is enabled, single conversions can be started by writing ADSC in ADCSRA register to one. ADSC can also be used to determine if a conversion is in progress. The ADSC bit will be read as one during a conversion, independently of how the conversion was started.

4.18.5 Prescaling and Conversion Timing

Figure 4-84. ADC Prescaler



By default, the successive approximation circuitry requires an input clock frequency between 50kHz and 200kHz to get maximum resolution. If a lower resolution than 10bits is needed, the input clock frequency to the ADC can be higher than 200kHz to get a higher sample rate.

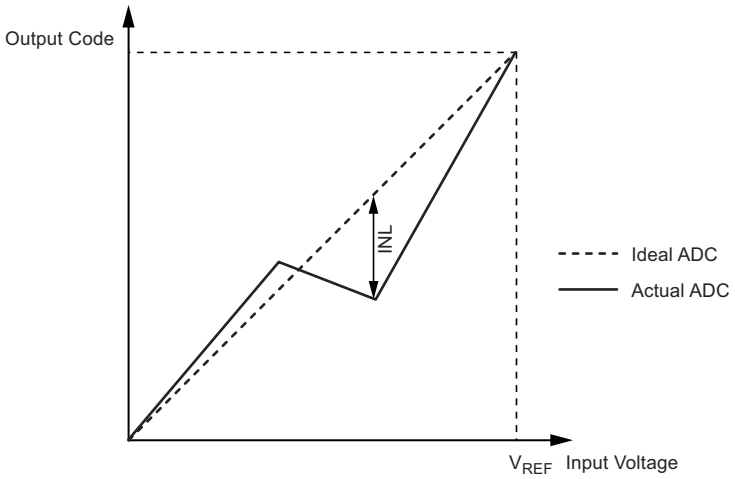
The ADC module contains a prescaler, which generates an acceptable ADC clock frequency from any CPU frequency above 100kHz. The prescaling is set by the ADPS bits in ADCSRA register. The prescaler starts counting from the moment the ADC is switched on by setting the ADEN bit in ADCSRA register. The prescaler keeps running for as long as the ADEN bit is set, and is continuously reset when ADEN is low.

When initiating a single ended conversion by setting the ADSC bit in ADCSRA register, the conversion starts at the following rising edge of the ADC clock cycle.

A normal conversion takes 13 ADC clock cycles. The first conversion after the ADC is switched on (ADEN in ADCSRA register is set) takes 25 ADC clock cycles in order to initialize the analog circuitry.

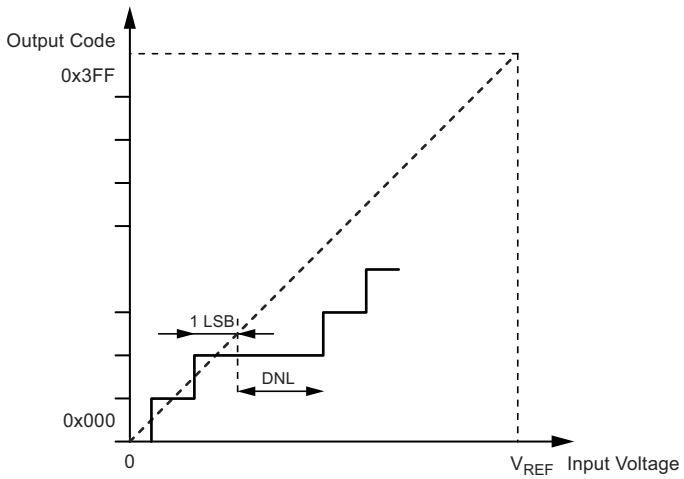
- Integral mon-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0LSB.

Figure 4-92. Integral Non-linearity (INL)



- Differential non-linearity (DNL): The maximum deviation of the actual code width (the interval between two adjacent transitions) from the ideal code width (1LSB). Ideal value: 0LSB.

Figure 4-93. Differential Non-linearity (DNL)



- Quantization error: Due to the quantization of the input voltage into a finite number of codes, a range of input voltages (1 LSB wide) will code to the same value. Always ±0.5LSB.
- Absolute accuracy: The maximum deviation of an actual (unadjusted) transition compared to an ideal transition for any code. This is the compound effect of offset, gain error, differential error, non-linearity, and quantization error. Ideal value: ±0.5LSB.

4.18.12.7 AMISCR – Analog Miscellaneous Control Register

Bit	7	6	5	4	3	2	1	0	
	-	-	-	-	-	**AREFEN**	**XREFEN**	ISRCEN	**AMISCR**
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:3 – Reserved Bits**

These bits are reserved for future use. For compatibility with future devices, they must be written to zero when AMISCR is written.

- **Bit 2 – AREFEN: External Voltage Reference Input Enable**

When this bit is written logic one, the voltage reference for the ADC is input from AREF pin as described in Table 4.18.11 on page 203. If active channels are used, using AV_{CC} or an external AREF higher than (AV_{CC} – 1V) is not recommended, as this will affect ADC accuracy. The internal voltage reference options may not be used if an external voltage is being applied to the AREF pin. It is recommended to use DIDR register bit function (digital input disable) when AREFEN is set.

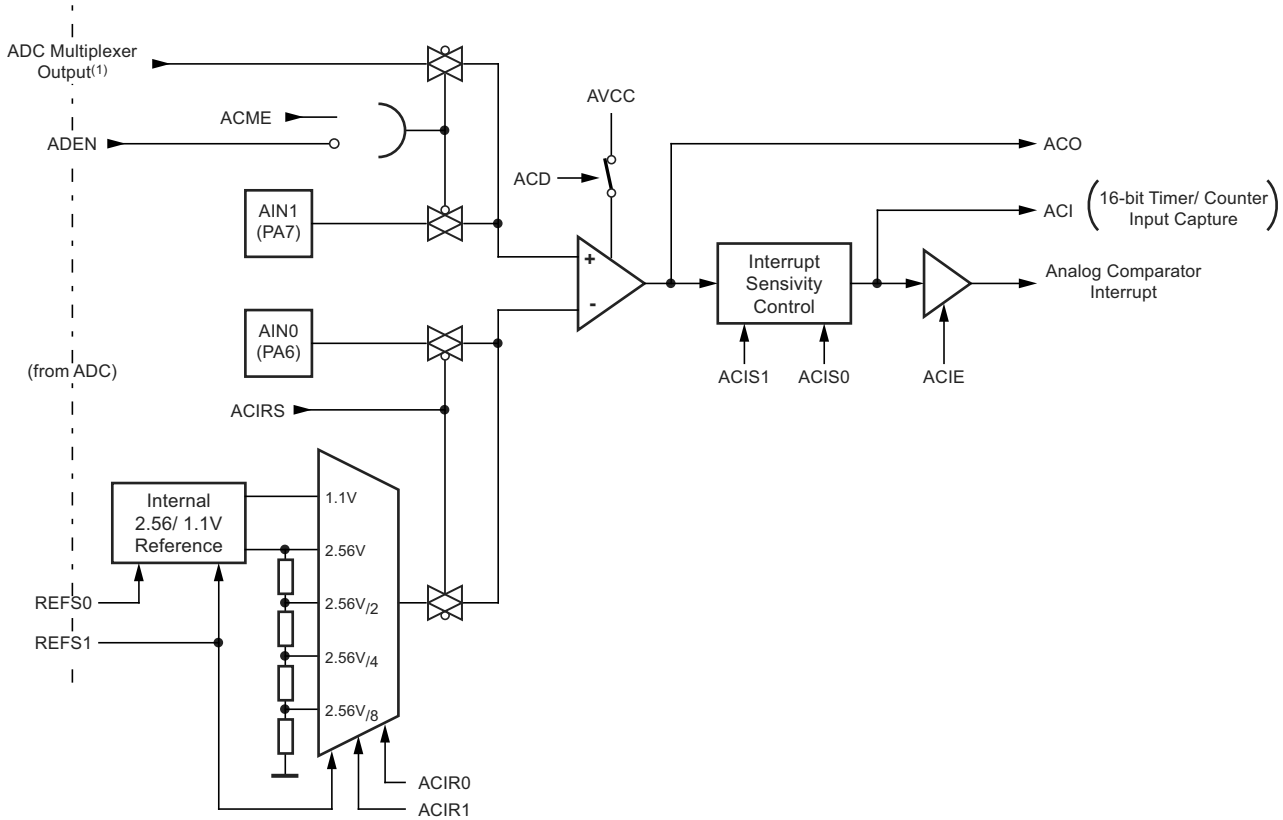
- **Bit 1 – XREFEN: Internal Voltage Reference Output Enable**

When this bit is written logic one, the internal voltage reference 1.1V or 2.56V is output on XREF pin as described in Table 4.18.11 on page 203. It is recommended to use DIDR register bit function (digital input disable) when XREFEN is set.

4.19 AnaComp - Analog Comparator

The analog comparator compares the input values on the positive pin (AIN1) and negative pin (AIN0). When the voltage on the positive pin is higher than the voltage on the negative pin, the analog comparator output, ACO, is set. The comparator can trigger a separate interrupt, exclusive to the analog comparator. The user can select Interrupt triggering on comparator output rise, fall or toggle. A block diagram of the comparator and its surrounding logic is shown in Figure 4-94.

Figure 4-94. Analog Comparator Block Diagram(1)(2)



Notes: 1. See Table 4-63 on page 212 and Table 4-64 on page 212
2. Refer to Table 4-24 on page 92 for Analog Comparator pin placement.

4.19.2 Analog Comparator Inputs

4.19.2.1 Analog Compare Positive Input

It is possible to select any of the inputs of the ADC positive input multiplexer to replace the positive input to the analog comparator. The ADC multiplexer is used to select this input, and consequently, the ADC must be switched off to utilize this feature. If the analog comparator multiplexer enable bit (ACME in ADCSRB register) is set and the ADC is switched off (ADEN in ADCSRA register is zero), MUX[4..0] in ADMUX register select the input pin to replace the positive input to the analog comparator, as shown in Table 4-63. If ACME is cleared or ADEN is set, AIN1 pin is applied to the positive input to the analog comparator.

Table 4-63. Analog Comparator Positive Input

ACME	ADEN	MUX[4..0]	Analog Comparator Positive Input - Comment	
0	x	x xxxx$_b$	AIN1	ADC switched on
x	1	x xxxx$_b$	AIN1	
1	0	0 0000$_b$	ADC0	ADC switched off
1	0	0 0001$_b$	ADC1	
1	0	0 0010$_b$	ADC2	
1	0	0 0011$_b$	ADC3 / ISRC	
1	0	0 0100$_b$	ADC4	
1	0	0 0101$_b$	ADC5	
1	0	0 0110$_b$	ADC6	
1	0	0 0111$_b$	ADC7	
1	0	0 1000$_b$	ADC8	
1	0	0 1001$_b$	ADC9	
1	0	0 1010$_b$	ADC10	
1	0	Other	This doesn't make sense - don't use	

4.19.2.2 Analog Compare Negative Input

It is possible to select an internal voltage reference to replace the negative input to the analog comparator. The output of a 2-bit DAC using the internal voltage reference of the DAC is available when ACIRS bit of ACSR register is set. The voltage reference division factor is done by ACIR[1..0] of ADCSRB register. If ACIRS is cleared, AIN0 pin is applied to the negative input to the analog comparator.

Table 4-64. Analog Comparator Negative Input

ACIRS	ACIR[1..0]	REFS[1..0]	Analog Comparator Negative Input - Comment
0	x	x	AIN0
1	x	0 0$_b$ 0 1$_b$ 1 0$_b$	Reserved
1	0 0$_b$	1 1$_b$	2.56V - using internal 2.56V voltage reference
1	0 1$_b$	1 1$_b$	1.28V ($^1/_2$ of 2.56V) - using internal 2.56V voltage reference
1	1 0$_b$	1 1$_b$	0.64V ($^1/_4$ of 2.56V - using internal 2.56V voltage reference
1	1 1$_b$	1 1$_b$	0.32V ($^1/_8$ of 2.56V) - using internal 2.56V voltage reference

4.21.2.4 Reading the Signature Row from Software

To read the signature row from software, load the Z-pointer with the signature byte address given in Table 4-65 on page 219 and set the SIGRD and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the SIGRD and SPMEN bits are set in SPMCSR, the signature byte value will be loaded in the destination register. The SIGRD and SPMEN bits will auto-clear upon completion of reading the signature row lock bits or if no LPM instruction is executed within three CPU cycles. When SIGRD and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Table 4-65. Signature Row Addressing

Signature Byte	Z-Pointer Address
Device signature byte 0	0x0000
Device signature byte 1	0x0002
Device signature byte 2	0x0004
8MHz RC oscillator calibration byte	0x0001
TSOFFSET - Temp sensor offset	0x0003
TSGAIN - Temp sensor gain	0x0005

Note: All other addresses are reserved for future use.

4.21.2.5 Preventing Flash Corruption

During periods of low V_{CC}, the flash program can be corrupted because the supply voltage is too low for the CPU and the flash to operate properly. These issues are the same as for board level systems using the flash, and the same design solutions should be applied.

A flash program corruption can be caused by two situations when the voltage is too low.

- First, a regular write sequence to the Flash requires a minimum voltage to operate correctly.
- Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage for executing instructions is too low.

Flash corruption can easily be avoided by following these design recommendations (one is sufficient):

1. Keep the AVR® RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal brown-out detector (BOD) if the operating voltage matches the detection level. If not, an external low V_{CC} reset protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.
2. Keep the AVR core in power-down sleep mode during periods of low V_{CC}. This will prevent the CPU from attempting to decode and execute instructions, effectively protecting the SPMCSR register and thus the flash from unintentional writes.

4.21.2.6 Programming Time for Flash when Using SPM

The calibrated RC Oscillator is used to time flash accesses. Table 4-66 shows the typical programming time for flash accesses from the CPU.

Table 4-66. SPM Programming Time

Symbol	Min Programming Time	Max Programming Time
Flash write (page erase, page write, and write lock bits by SPM)	3.7ms	4.5ms

4.25.5 Pin Pull-up

Figure 4-124. I/O Pin Pull-up Resistor Current versus Input Voltage (V_{CC} = 2.7V)

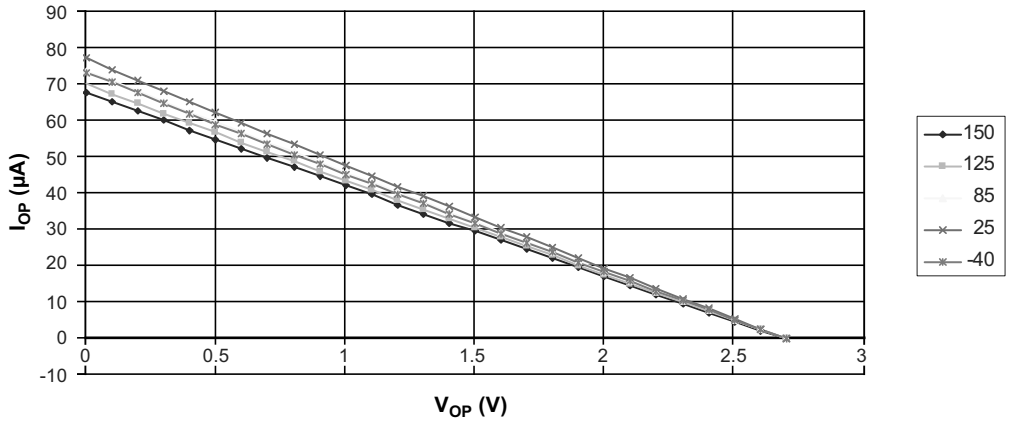


Figure 4-125. I/O Pin pull-up Resistor Current versus Input Voltage (V_{CC} = 5V)

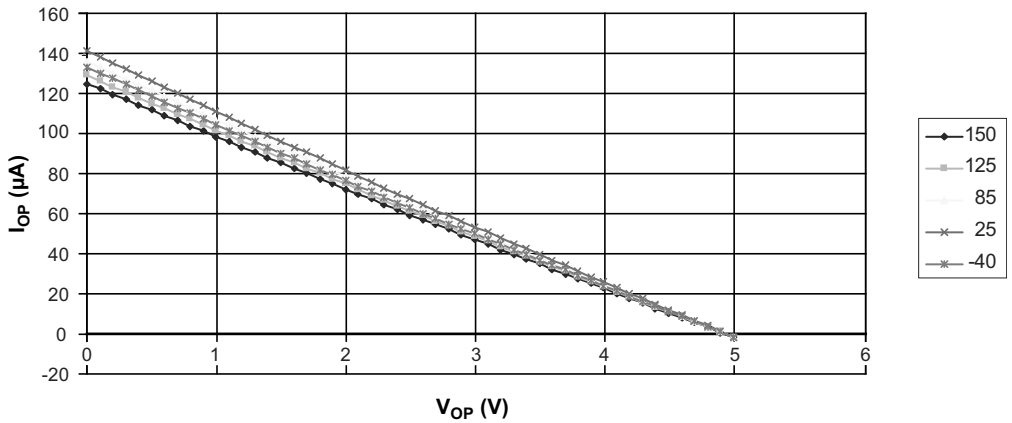


Figure 4-126. Reset Pull-up Resistor Current versus Reset Pin Voltage (V_{CC} = 2.7V)