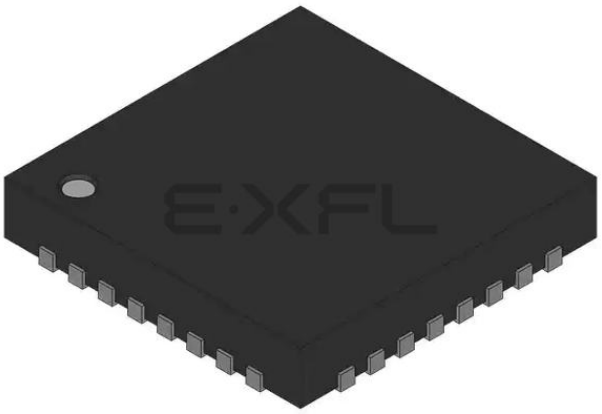
Atmel - ATMEGA64C1-15MD Datasheet



Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

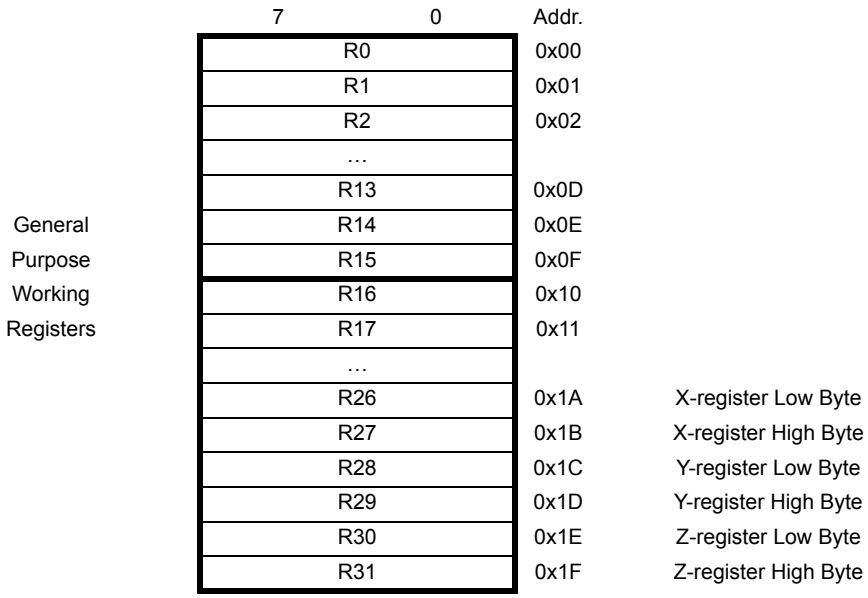
"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" 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	Active
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	CANbus, LINbus, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, Temp Sensor, WDT
Number of I/O	-
Program Memory Size	64KB (32K x 16)
Program Memory Type	FLASH
EEPROM Size	2K x 8
RAM Size	4K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 5.5V
Data Converters	A/D 11x10b; D/A 1x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 150°C (TA)
Mounting Type	Surface Mount
Package / Case	32-VQFN Exposed Pad
Supplier Device Package	32-QFN (7x7)
Purchase URL	https://www.e-xfl.com/product-detail/atmel/atmega64c1-15md

Figure 3-2 shows the structure of the 32 general purpose working registers in the CPU.

Figure 3-2. AVR CPU General Purpose Working Registers



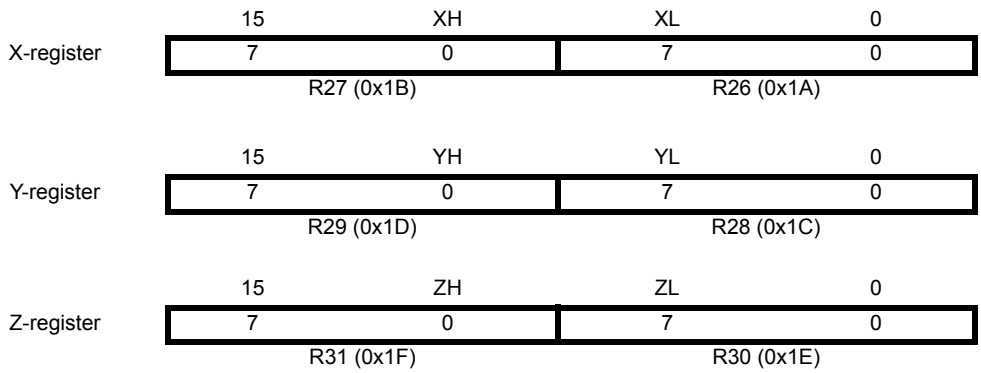
Most of the instructions operating on the register file have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 3-2, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user data space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

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

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

Figure 3-3. The X-, Y-, and Z-registers

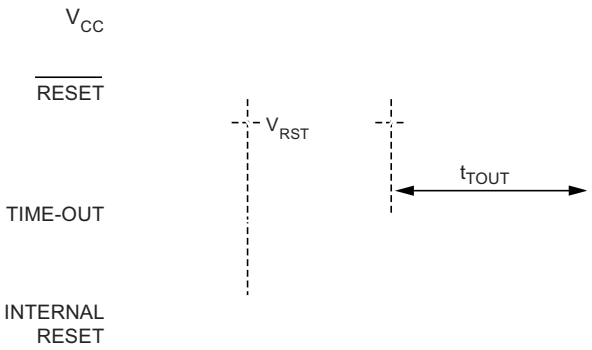


In the different addressing modes these address registers have functions as fixed displacement, automatic increment, and automatic decrement (see the instruction set reference for details).

7.2.2 External Reset

An external reset is generated by a low level on the $\overline{\text{RESET}}$ pin. Reset pulses longer than the minimum pulse width (see Table 7-1) 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 counter starts the MCU after the Time-out period – t_{TOUT} – has expired.

Figure 7-4. External Reset during Operation



7.2.3 Brown-out Detection

ATmega16/32/64/M1/C1 has an on-chip brown-out detection (BOD) circuit for monitoring the V_{CC} level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL Fuses. The trigger level has a hysteresis to ensure spike free brown-out detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} – V_{HYST}/2$.

Table 7-2. BODLEVEL Fuse Coding(1)(2)

BODLEVEL 2..0 Fuses	Typ V_{BOT}	Unit
111	Disabled	
110	4.5	V
011	4.4	V
100	4.3	V
010	4.2	V
001	2.8	V
101	2.7	V
000	2.6	V

Notes: 1. V_{BOT} may be below nominal minimum operating voltage for some devices. For devices where this is the case, the device is tested down to $V_{CC} = V_{BOT}$ during the production test. This guarantees that a brown-out reset will occur before V_{CC} drops to a voltage where correct operation of the microcontroller is no longer guaranteed. The test is performed using BODLEVEL = 010 for low operating voltage and BODLEVEL = 101 for high operating voltage.

2. Values are guidelines only.

Table 7-3. Brown-out Characteristics(1)

Parameter	Symbol	Min.	Typ.	Max.	Unit
Brown-out Detector Hysteresis	V_{HYST}		80		mV
Min Pulse Width on Brown-out Reset	t_{BOD}		2		µs

Note: 1. Values are guidelines only.

9.4.5 Port C Data Direction Register – DDRC

Bit	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
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	

9.4.6 Port C Input Pins Address – PINC

Bit	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	PINC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

9.4.7 Port D Data Register – PORTD

Bit	7	6	5	4	3	2	1	0	
	PORTD7	PORTD6	PORTD5	PORTD4	PORTD3	PORTD2	PORTD1	PORTD0	PORTD
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	

9.4.8 Port D Data Direction Register – DDRD

Bit	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
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	

9.4.9 Port D Input Pins Address – PIND

Bit	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	PIND
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

9.4.10 Port E Data Register – PORTE

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	PORTE2	PORTE1	PORTE0	PORTE
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

9.4.11 Port E Data Direction Register – DDRE

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	DDE2	DDE1	DDE0	DDRE
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

9.4.12 Port E Input Pins Address – PINE

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	PINE2	PINE1	PINE0	PINE
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	N/A	N/A	N/A	

Table 12-6 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to fast PWM mode.

Table 12-6. Compare Output Mode, Fast PWM Mode(1)

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on compare match, set OC0B at TOP
1	1	Set OC0B on compare match, clear OC0B at TOP

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 12.6.3 “Fast PWM Mode” on page 83 for more details.

Table 12-7 shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 12-7. Compare Output Mode, Phase Correct PWM Mode(1)

COM0B1	COM0B0	Description
0	0	Normal port operation, OC0B disconnected.
0	1	Reserved
1	0	Clear OC0B on compare match when up-counting. Set OC0B on compare match when down-counting.
1	1	Set OC0B on compare match when up-counting. Clear OC0B on compare match when down-counting.

Note: 1. A special case occurs when OCR0B equals TOP and COM0B1 is set. In this case, the compare match is ignored, but the set or clear is done at TOP. See Section 12.6.4 “Phase Correct PWM Mode” on page 84 for more details.

- **Bits 3, 2 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16/32/64/M1/C1 and will always read as zero.

- **Bits 1:0 – WGM01:0: Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used, see Table 12-8. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), clear timer on compare match (CTC) mode, and two types of pulse width modulation (PWM) modes (see Section 12.6 “Modes of Operation” on page 81).

Table 12-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on(1)(2)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, phase correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, phase correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	TOP	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

12.8.3 Timer/Counter Register – TCNT0

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
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 Timer/Counter register gives direct access, both for read and write operations, to the Timer/Counter unit 8-bit counter. Writing to the TCNT0 register blocks (removes) the compare match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a compare match between TCNT0 and the OCR0x registers.

12.8.4 Output Compare Register A – OCR0A

Bit	7	6	5	4	3	2	1	0	
	OCR0A[7:0]								OCR0A
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 register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0A pin.

12.8.5 Output Compare Register B – OCR0B

Bit	7	6	5	4	3	2	1	0	
	OCR0B[7:0]								OCR0B
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 register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an output compare interrupt, or to generate a waveform output on the OC0B pin.

12.8.6 Timer/Counter Interrupt Mask Register – TIMSK0

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	OCIE0B	OCIE0A	TOIE0	TIMSK0
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 – Res: Reserved Bits**

These bits are reserved bits in the ATmega16/32/64/M1/C1 and will always read as zero.

- **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the status register is set, the Timer/Counter compare match B interrupt is enabled. The corresponding interrupt is executed if a compare match in Timer/Counter occurs, i.e., when the OCF0B bit is set in the Timer/Counter interrupt flag register – TIFR0.

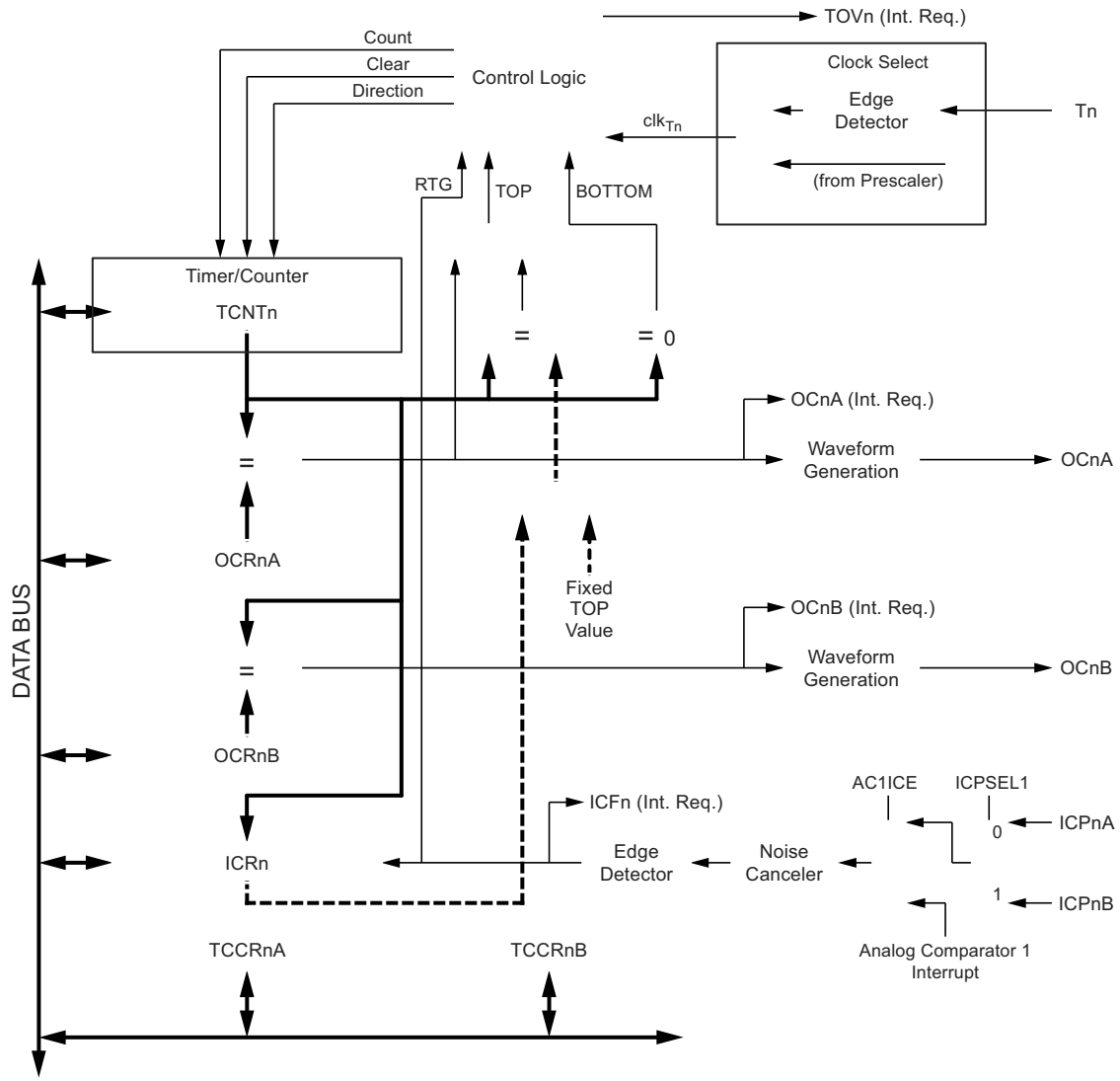
- **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**

When the OCIE0A bit is written to one, and the I-bit in the status register is set, 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/Counter 0 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, 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/Counter 0 interrupt flag register – TIFR0.

Figure 13-1. 16-bit Timer/Counter Block Diagram(1)



Note: 1. Refer to Table on page 5 for Timer/Counter 1 pin placement and description.

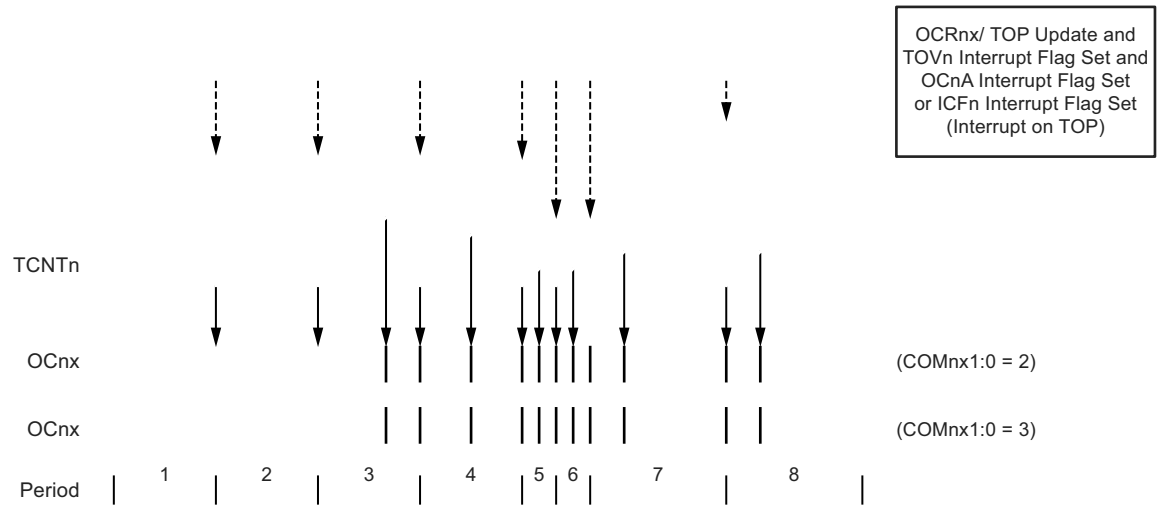
13.1.1 Registers

The Timer/Counter (TCNTn), output compare registers (OCRnx), and input capture register (ICRn) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in Section 13.2 “Accessing 16-bit Registers” on page 94. The Timer/Counter control registers (TCCRnx) are 8-bit registers and have no CPU access restrictions. Interrupt requests (abbreviated to Int.Req. in the figure) signals are all visible in the timer interrupt flag register (TIFRn). All interrupts are individually masked with the timer interrupt mask register (TIMSKn). TIFRn and TIMSKn are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the Tn pin. The clock select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk_{Tn}).

The double buffered output compare registers (OCRnx) are compared with the Timer/Counter value at all time. The result of the compare can be used by the waveform generator to generate a PWM or variable frequency output on the output compare pin (OCnx). See Section 13.6 “Output Compare Units” on page 99 The compare match event will also set the compare match flag (OCFnx) which can be used to generate an output compare interrupt request.

Figure 13-7. Fast PWM Mode, Timing Diagram



The Timer/Counter overflow flag (TOVn) is set each time the counter reaches TOP. In addition the OCnA or ICFn Flag is set at the same timer clock cycle as TOVn is set when either OCRnA or ICRn is used for defining the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be used for updating the TOP and compare values.

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 TCNTn and the OCRnx. Note that when using fixed TOP values the unused bits are masked to zero when any of the OCRnx registers are written.

The procedure for updating ICRn differs from updating OCRnA when used for defining the TOP value. The ICRn register is not double buffered. This means that if ICRn is changed to a low value when the counter is running with none or a low prescaler value, there is a risk that the new ICRn value written is lower than the current value of TCNTn. The result will then be that the counter will miss the compare match at the TOP value. The counter will then have to count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCRnA Register however, is double buffered.

This feature allows the OCRnA I/O location to be written anytime. When the OCRnA I/O location is written the value written will be put into the OCRnA buffer register.

The OCRnA compare register will then be updated with the value in the buffer register at the next timer clock cycle the TCNTn matches TOP. The update is done at the same timer clock cycle as the TCNTn is cleared and the TOVn flag is set.

Using the ICRn register for defining TOP works well when using fixed TOP values. By using ICRn, the OCRnA register is free to be used for generating a PWM output on OCnA. However, if the base PWM frequency is actively changed (by changing the TOP value), using the OCRnA as TOP is clearly a better choice due to its double buffer feature.

In fast PWM mode, the compare units allow generation of PWM waveforms on the OCnx pins. Setting the COMnx1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COMnx1:0 to three (see Table on page 110). The actual OCnx value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OCnx). The PWM waveform is generated by setting (or clearing) the OCnx Register at the compare match between OCRnx and TCNTn, and clearing (or setting) the OCnx Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output can be calculated by the following equation:

$$f_{OCnxPWM} = \frac{f_{\text{clk\_I/O}}}{N \cdot (1 + TOP)}$$

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

The extreme values for the OCRnx Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCRnx is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCRnx equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COMnx1:0 bits.)

Figure 13-12 shows the count sequence close to TOP in various modes. When using phase and frequency correct PWM mode the OCRnx register is updated at BOTTOM. The timing diagrams will be the same, but TOP should be replaced by BOTTOM, TOP-1 by BOTTOM+1 and so on. The same renaming applies for modes that set the TOVn Flag at BOTTOM.

Figure 13-12.Timer/Counter Timing Diagram, no Prescaling

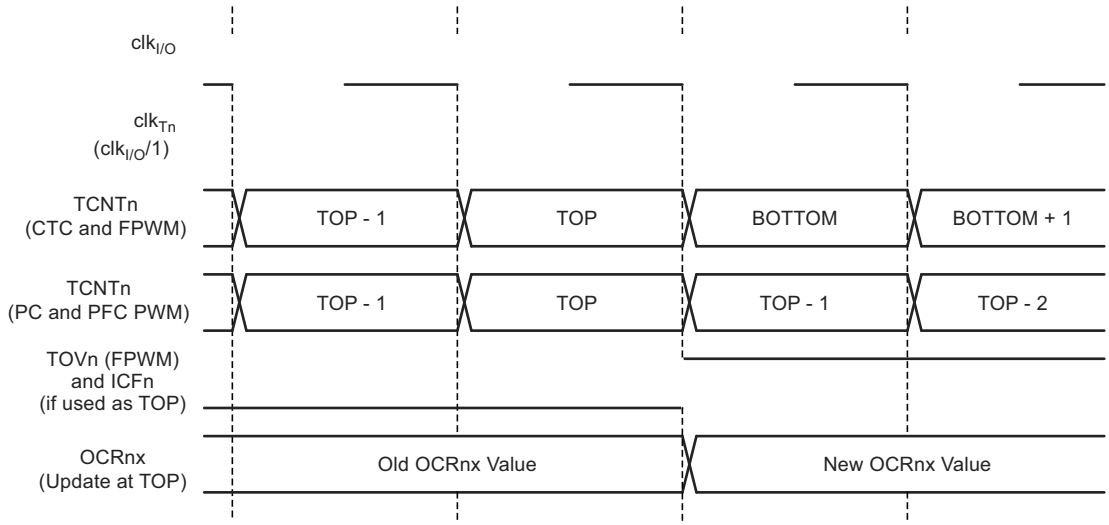


Figure 13-13 shows the same timing data, but with the prescaler enabled.

Figure 13-13.Timer/Counter Timing Diagram, with Prescaler ($f_{clk\_I/O}$/8)

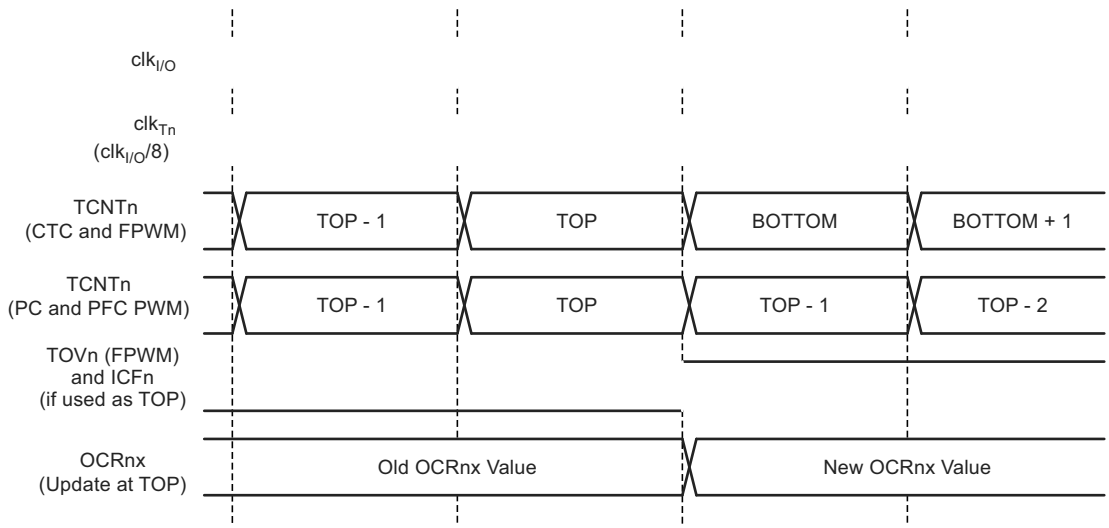


Table 14-9. Synchronization Source Description in Centered Mode

PSYNCn1	PSYNCn0	Description
0	0	Send signal on match with OCRnRA (during counting down of PSC). The min value of OCRnRA must be 1.
0	1	Send signal on match with OCRnRA (during counting up of PSC). The min value of OCRnRA must be 1.
1	0	no synchronization signal
1	1	no synchronization signal

14.16.3 PSC Output Compare SA Register – POCRnSAH and POCRnSAL

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	POCRnSA[11:8]				POCRnSAH
	POCRnSA[7:0]								POCRnSAL
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	

14.16.4 PSC Output Compare RA Register – POCRnRAH and POCRnRAL

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	POCRnRA[11:8]				POCRnRAH
	POCRnRA[7:0]								POCRnRAL
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	

14.16.5 PSCOutput Compare SB Register – POCRnSBH and POCRnSBL

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	POCRnSB[11:8]				POCRnSBH
	POCRnSB[7:0]								OCRnSBL
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	

14.16.6 PSC Output Compare RB Register – POCR_RBH and POCR_RBL

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	POCRnRB[11:8]				POCR_RBH
	POCRnRB[7:0]								POCR_RBL
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	

Note: n = 0 to 2 according to module number.

The output compare registers RA, RB, SA and SB contain a 12-bit value that is continuously compared with the PSC counter value. A match can be used to generate an output compare interrupt, or to generate a waveform output on the associated pin.

The output compare registers are 16bit and 12-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.

16.10.7 CAN Status Interrupt MOb Registers - CANSIT2 and CANSIT1

Bit	7	6	5	4	3	2	1	0	
	-	-	SIT5	SIT4	SIT3	SIT2	SIT1	SIT0	CANSIT2
	-	-	-	-	-	-	-	-	CANSIT1
Bit	15	14	13	12	11	10	9	8	
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 5:0 - SIT5:0: Status of Interrupt by MOb**
  - 0 - no interrupt.
  - 1- MOb interrupt.

Note: Example: CANSIT2 = 0010 0001_b: MOb 0 and 5 interrupts.

- **Bit 15:6 – Reserved Bits**

These bits are reserved for future use.

16.10.8 CAN Bit Timing Register 1 - CANBT1

Bit	7	6	5	4	3	2	1	0	
	-	BRP5	BRP4	BRP3	BRP2	BRP1	BRP0	-	CANBT1
Read/Write	-	R/W	R/W	R/W	R/W	R/W	R/W	-	
Initial Value	-	0	0	0	0	0	0	-	

- **Bit 7– Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT1 is written.

- **Bit 6:1 – BRP5:0: Baud Rate Prescaler**

The period of the CAN controller system clock Tscl is programmable and determines the individual bit timing.

$$\text{Tscl} = \frac{\text{BRP[5:0]} + 1}{\text{clk}_{\text{IO}}\text{frequency}}$$

If ‘BRP[5..0]=0’, see Section 16.4.3 “Baud Rate” on page 148 and Section • “Bit 0 – SMP: Sample Point(s)” on page 164.

- **Bit 0 – Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT1 is written.

16.10.9 CAN Bit Timing Register 2 - CANBT2

Bit	7	6	5	4	3	2	1	0	
	-	SJW1	SJW0	-	PRS2	PRS1	PRS0	-	CANBT2
Read/Write	-	R/W	R/W	-	R/W	R/W	R/W	-	
Initial Value	-	0	0	-	0	0	0	-	

- **Bit 7– Reserved Bit**

This bit is reserved for future use. For compatibility with future devices, it must be written to zero when CANBT2 is written.

17. LIN / UART - Local Interconnect Network Controller or UART

The LIN (Local Interconnect Network) is a serial communications protocol which efficiently supports the control of mechatronics nodes in distributed automotive applications. The main properties of the LIN bus are:

- Single master with multiple slaves concept
- Low cost silicon implementation based on common UART/SCI interface
- Self synchronization in slave node
- Deterministic signal transmission with signal propagation time computable in advance
- Low cost single-wire implementation
- Speed up to 20Kbit/s.

LIN provides a cost efficient bus communication where the bandwidth and versatility of CAN are not required. The specification of the line driver/receiver needs to match the ISO9141 NRZ-standard.

If LIN is not required, the controller alternatively can be programmed as universal asynchronous serial receiver and transmitter (UART).

17.1 LIN Features

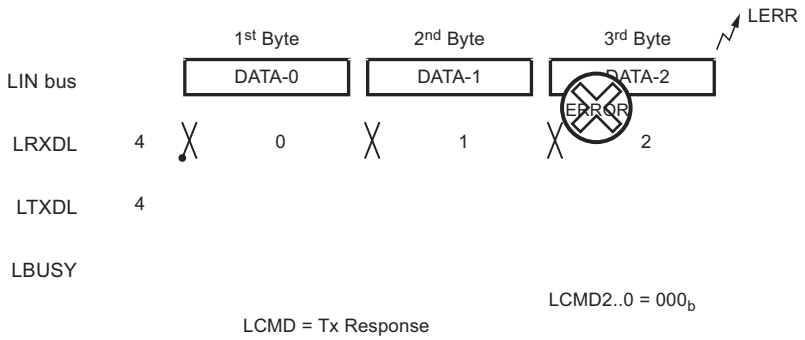
- Hardware implementation of LIN 2.1 (LIN 1.3 compatibility)
- Small, CPU efficient and independent master/slave routines based on “LIN Work Flow Concept” of LIN 2.1 specification
- Automatic LIN header handling and filtering of irrelevant LIN frames
- Automatic LIN response handling
- Extended LIN error detection and signaling
- Hardware frame time-out detection
- “Break-in-data” support capability
- Automatic re-synchronization to ensure proper frame integrity
- Fully flexible extended frames support capabilities

17.2 UART Features

- Full duplex operation (independent serial receive and transmit processes)
- Asynchronous operation
- High resolution baud rate generator
- Hardware support of 8 data bits, odd/even/no parity bit, 1 stop bit frames
- Data over-run and framing error detection

17.5.7.5 Data Length after Error

Figure 17-11. Tx Response - Error



Note: Information on response (ex: error on byte) is only available at the end of the serialization/de-serialization of the byte.

17.5.7.6 Data Length in UART Mode

- The UART mode forces LRXDL and LTXDL to 0 and disables the writing in LINDLR register,
- Note that after reset, LRXDL and LTXDL are also forced to 0.

17.5.8 xxOK Flags

There are three xxOK flags in LINSIR register:

- LIDOK: LIN IDentifier OK
  It is set at the end of the header, either by the Tx header function or by the Rx header. In LIN 1.3, before generating LIDOK, the controller updates the LRXDL and LTXDL fields in LINDLR register.
  It is not driven in UART mode.
- LRXOK: LIN RX response complete
  It is set at the end of the response by the Rx response function in LIN mode and once a character is received in UART mode.
- LTXOK: LIN TX response complete
  It is set at the end of the response by the Tx Response function in LIN mode and once a character has been sent in UART mode.

These flags can generate interrupts if the corresponding enable interrupt bit is set in the LINENIR register (see Section 17.5.13 “Interrupts” on page 188).

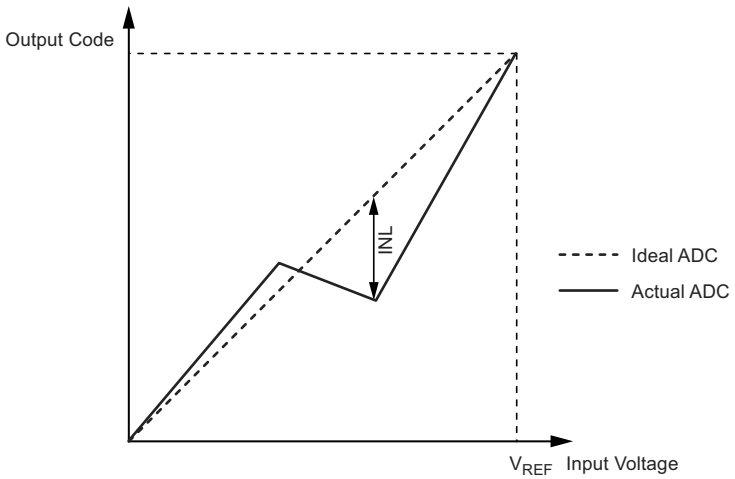
17.5.9 xxERR Flags

LERR bit of the LINSIR register is an logical ‘OR’ of all the bits of LINERR register (see Section 17.5.13 “Interrupts” on page 188). There are eight flags:

- LBERR = LIN Bit ERRor.
  A unit that is sending a bit on the bus also monitors the bus. A LIN bit error will be flagged when the bit value that is monitored is different from the bit value that is sent. After detection of a LIN bit error the transmission is aborted.
- LCERR = LIN Checksum ERRor.
  A LIN checksum error will be flagged if the inverted modulo-256 sum of all received data bytes (and the protected identifier in LIN 2.1) added to the checksum does not result in 0xFF.

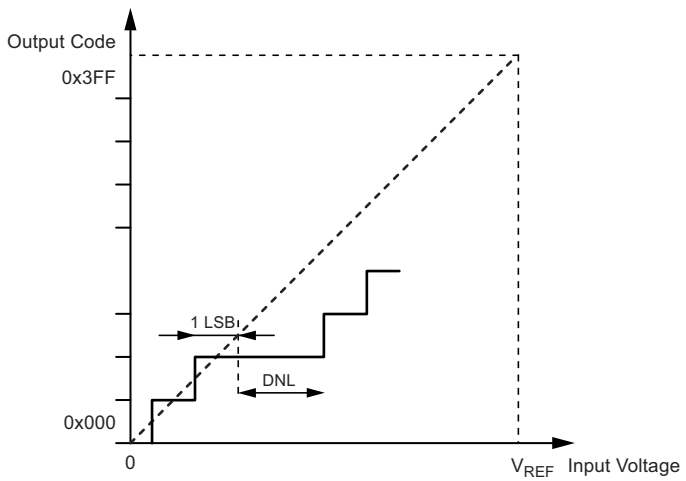
- Integral non-linearity (INL): After adjusting for offset and gain error, the INL is the maximum deviation of an actual transition compared to an ideal transition for any code. Ideal value: 0 LSB.

Figure 18-11. 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 (1 LSB). Ideal value: 0 LSB.

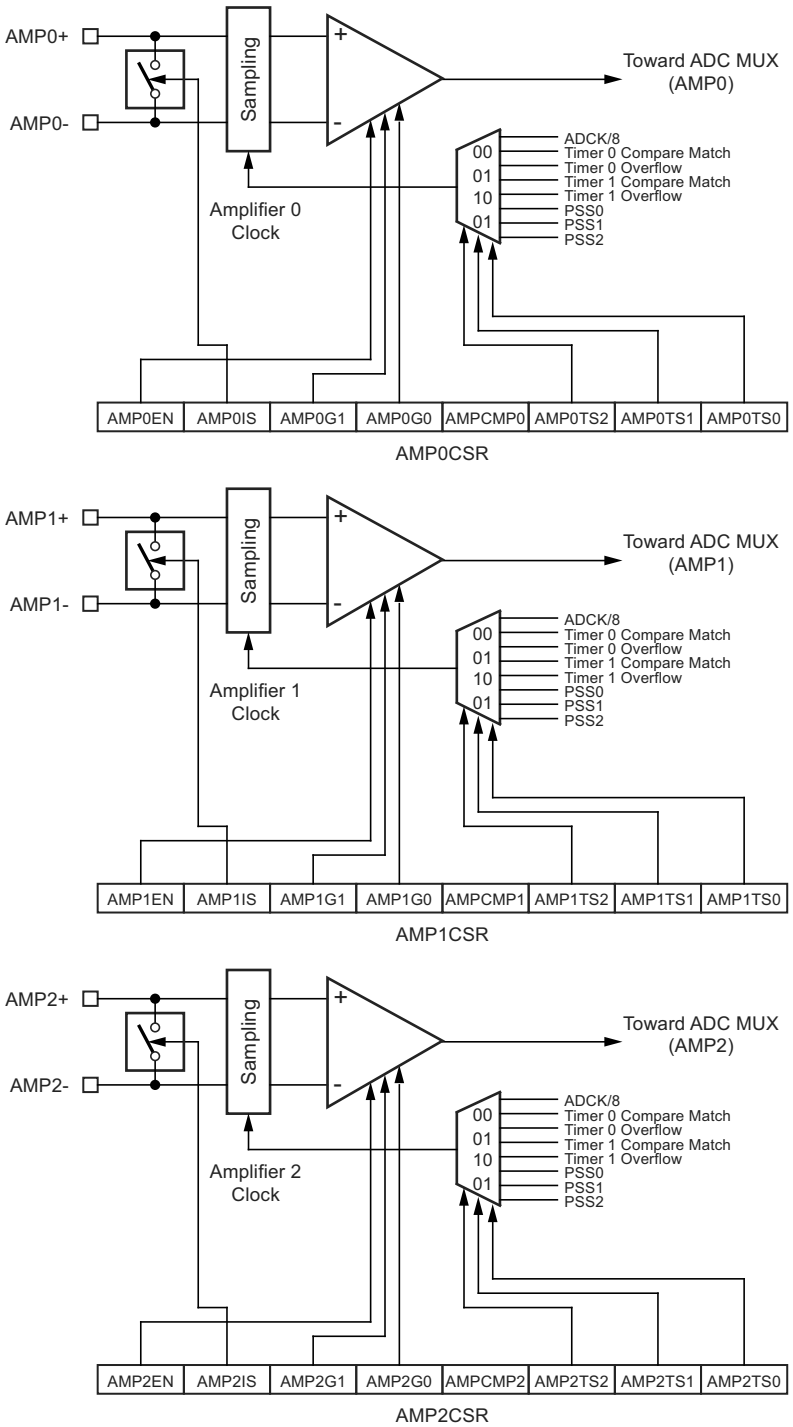
Figure 18-12. Differential Non-linearity (DNL)



- 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 ±0.5 LSB.
- 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: ±0.5 LSB.

The block diagram of the two amplifiers is shown on Figure 18-17.

Figure 18-17. Amplifiers Block Diagram



18.11 Amplifier Control Registers

The configuration of the amplifiers are controlled via two dedicated registers AMP0CSR and AMP1CSR. Then the start of conversion is done via the ADC control and status registers.

The conversion result is stored on ADCH and ADCL register which contain respectively the most significant bits and the less significant bits.

18.11.1 Amplifier 0 control and status register – AMP0CSR

Bit	7	6	5	4	3	2	1	0	
	AMP0EN	**AMP0IS**	**AMP0G1**	**AMP0G0**	**AMPCMP0**	**AMP0TS2**	**AMP0TS1**	**AMP0TS0**	**AMP0CSR**
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 – AMP0EN: Amplifier 0 Enable Bit**

Set this bit to enable the amplifier 0.
Clear this bit to disable the amplifier 0.
Clearing this bit while a conversion is running will take effect at the end of the conversion.

Warning: Always clear AMP0TS0:1 when clearing AMP0EN.

- **Bit 6 – AMP0IS: Amplifier 0 Input Shunt**

Set this bit to short-circuit the amplifier 0 input.
Clear this bit to normally use the amplifier 0.

- **Bit 5, 4 – AMP0G1, 0: Amplifier 0 Gain Selection Bits**

These 2 bits determine the gain of the amplifier 0.
The different setting are shown in Table 18-8.

Table 18-8. Amplifier 0 Gain Selection

AMP0G1	AMP0G0	Description
0	0	Gain 5
0	1	Gain 10
1	0	Gain 20
1	1	Gain 40

To ensure an accurate result, after the gain value has been changed, the amplifier input needs to have a quite stable input value during at least 4 Amplifier synchronization clock periods.

- **Bit 3 – AMPCMP0: Amplifier 0 - Comparator 0 Connection**

Set this bit to connect the amplifier 0 to the comparator 0 positive input. In this configuration the comparator clock is twice the amplifier clock. Clear this bit to normally use the Amplifier 0.

- **Bit 2:0 – AMP0TS2,AMP0TS1,AMP0TS0: Amplifier 0 Clock Source Selection Bits**

In accordance with Table 18-9 on page 219, these 3 bits select the event which will generate the clock for the amplifier 0. This clock source is necessary to start the conversion on the amplified channel.

25.2 Fuse Bits

The ATmega16/32/64/M1/C1 has three Fuse bytes. Table 25-4 to Table 25-7 on page 257 describe briefly the functionality of all the fuses and how they are mapped into the Fuse bytes. Note that the fuses are read as logical zero, “0”, if they are programmed.

Table 25-4. Extended Fuse Byte

Extended Fuse Byte	Bit No	Description	Default Value
-	7	-	1 (unprogrammed)
-	6	-	1 (unprogrammed)
PSCRB	5	PSC reset behavior	1 (unprogrammed)
PSCRVA	4	PSCOUTnA reset value	1 (unprogrammed)
PSCRVB	3	PSCOUTnB reset value	1 (unprogrammed)
BODLEVEL2(1)	2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL1(1)	1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL0(1)	0	Brown-out detector trigger level	1 (unprogrammed)

Note: 1. See Table 7-2 on page 40 for BODLEVEL fuse decoding.

25.3 PSC Output Behavior during Reset

For external component safety reason, the state of PSC outputs during reset can be programmed by fuses PSCRB, PSCARV and PSCBRV. These fuses are located in the extended fuse byte (see Table 25-4 on page 256).

If PSCRB fuse equals 1 (unprogrammed), all PSC outputs keep a standard port behavior. If PSC0RB fuse equals 0 (programmed), all PSC outputs are forced at reset to low level or high level according to PSCARV and PSCBRV fuse bits. In this second case, the PSC outputs keep the forced state until POC register is written. Section 5.10.1 “Clock Prescaler Register – CLKPR” on page 33

PSCARV (PSCOUTnA reset value) gives the state low or high which will be forced on PSCOUT0A, PSCOUT1A and PSCOUT2A outputs when PSCRB is programmed. If PSCARV fuse equals 0 (programmed), the PSCOUT0A, PSCOUT1A and PSCOUT2A outputs will be forced to high state. If PSCRV fuse equals 1 (unprogrammed), the PSCOUT0A, PSCOUT1A and PSCOUT2A outputs will be forced to low state.

PSCBRV (PSCOUTnB Reset Value) gives the state low or high which will be forced on PSCOUT0B, PSCOUT1B and PSCOUT2B outputs when PSCRB is programmed. If PSCBRV fuse equals 0 (programmed), the PSCOUT0B, PSCOUT1B and PSCOUT2B outputs will be forced to high state. If PSCRV fuse equals 1 (unprogrammed), the PSCOUT0B, PSCOUT1B and PSCOUT2B outputs will be forced to low state.

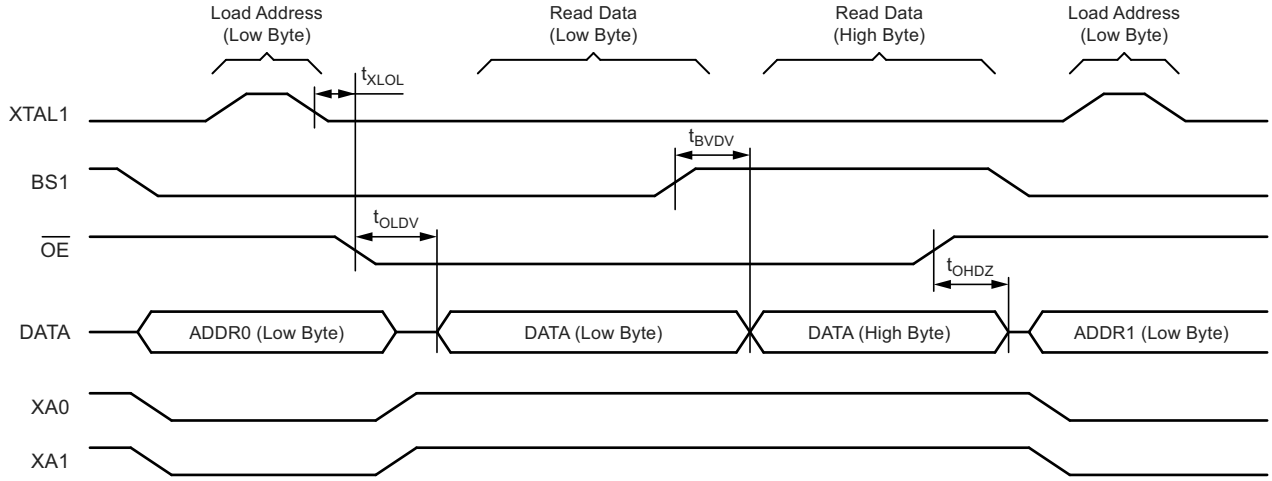
Table 25-5. PSC Output Behavior during and after Reset until POC Register is Written

PSCRB	PSCARV	PSCBRV	PSCOUTnA	PSCOUTnB
Unprogrammed	X	X	Normal port	Normal port
Programmed	Unprogrammed	Unprogrammed	Forced low	Forced low
Programmed	Unprogrammed	Programmed	Forced low	Forced high
Programmed	Programmed	Unprogrammed	Forced high	Forced low
Programmed	Programmed	Programmed	Forced high	Forced high
BODLEVEL2(1)		2	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL1(1)		1	Brown-out detector trigger level	1 (unprogrammed)
BODLEVEL0(1)		0	Brown-out detector trigger level	1 (unprogrammed)

Table 26-7. ADC Characteristics in Differential Mode - T_A = –40°C to +125°C, V_{CC} = 2.7V to 5.5V (unless otherwise noted)

Parameter	Condition	Symbol	Min	Typ	Max	Unit
Resolution	Differential conversion, gain = 5x			8		Bits
	Differential conversion, gain = 10x			8		
	Differential conversion, gain = 20x			8		
	Differential conversion, gain = 40x			8		
Absolute accuracy	Gain = 5x, 10x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz	TUE		1.5	3.5	LSB
	Gain = 20x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			1.5	4.0	
	Gain = 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			1.5	4.5	
Integral non-linearity	Gain = 5x, 10x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz	INL		0.1	1.5	LSB
	Gain = 20x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			0.2	2.5	
	Gain = 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 1MHz			0.3	3.5	
	Gain = 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			0.7	4.5	
Differential non-linearity	Gain = 5x, 10x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz	DNL		0.1	1.0	LSB
	Gain = 20x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			0.2	1.5	
	Gain = 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz			0.3	2.5	
Gain error	Gain = 5x, 10x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz		–3.0		+3.0	LSB
	Gain = 20x, 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz		–3.0		+3.0	
Offset error	Gain = 5x, 10x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz		–3.0		+3.0	LSB
	Gain = 20x, 40x, V_{CC} = 5V, V_{REF} = 2.56V, ADC clock = 2MHz		–4.0		+4.0	
Ref voltage		V_{REF}	2.56		AVCC – 0.5	V

Figure 26-7. Parallel Programming Timing, Reading Sequence (within the Same Page) with Timing Requirements(1)



Note: 1. The timing requirements shown in Figure 25-7 on page 268 (i.e., t_{DVXH}, t_{XHXL}, and t_{XLDX}) also apply to reading operation.

Table 26-8. Parallel Programming Characteristics, V_{CC} = 5V ±10%

Parameter	Symbol	Min.	Typ.	Max.	Unit
Programming enable voltage	V_{PP}	11.5		12.5	V
Programming enable current	I_{PP}			250	µA
Data and control valid before XTAL1 high	t_{DVXH}	67			ns
XTAL1 low to XTAL1 high	t_{XLXH}	200			ns
XTAL1 pulse width high	t_{XHXL}	150			ns
Data and control hold after XTAL1 low	t_{XLDX}	67			ns
XTAL1 low to $\overline{WR}$ low	t_{XLWL}	0			ns
XTAL1 low to PAGEL high	t_{XLPH}	0			ns
PAGEL low to XTAL1 high	t_{PLXH}	150			ns
BS1 valid before PAGEL high	t_{BVPH}	67			ns
PAGEL pulse width high	t_{PHPL}	150			ns
BS1 hold after PAGEL low	t_{PLBX}	67			ns
BS2/1 hold after $\overline{WR}$ low	t_{WLBX}	67			ns
PAGEL low to $\overline{WR}$ low	t_{PLWL}	67			ns
BS1 valid to $\overline{WR}$ low	t_{BVWL}	67			ns
$\overline{WR}$ pulse width low	t_{WLWH}	150			ns
$\overline{WR}$ low to RDY/$\overline{BSY}$ low	t_{WLRL}	0		1	µs
$\overline{WR}$ low to RDY/$\overline{BSY}$ high(1)	t_{WLRH}	3.7		5	ms
$\overline{WR}$ low to RDY/$\overline{BSY}$ high for chip erase(2)	t_{WLRH_CE}	7.5		10	ms
XTAL1 low to $\overline{OE}$ low	t_{XLOL}	0			ns
BS1 valid to DATA valid	t_{BVDV}	0		250	ns
$\overline{OE}$ low to DATA valid	t_{OLDV}			250	ns
$\overline{OE}$ high to DATA tri-stated	t_{OHDZ}			250	ns

Notes: 1. t_{WLRH} is valid for the write flash, write EEPROM, write fuse bits and write lock bits commands.
2. t_{WLRH_CE} is valid for the chip erase command.

29. Register Summary (Continued)

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
0x33 (0x53)	SMCR	–	–	–	–	SM2	SM1	SM0	SE	34
0x32 (0x52)	MSMCR	Monitor Stop Mode Control Register								Reserved
0x31 (0x51)	MONDR	Monitor Data Register								Reserved
0x30 (0x50)	ACSR	AC3IF	AC2IF	AC1IF	AC0IF	AC3O	AC2O	AC1O	AC0O	231
0x2F (0x4F)	Reserved	–	–	–	–	–	–	–	–	
0x2E (0x4E)	SPDR	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0	139
0x2D (0x4D)	SPSR	SPIF	WCOL	–	–	–	–	–	SPI2X	139
0x2C (0x4C)	SPCR	SPIE	SPE	DORD	MSTR	CPOL	CPHA	SPR1	SPR0	138
0x2B (0x4B)	Reserved	–	–	–	–	–	–	–	–	
0x2A (0x4A)	Reserved	–	–	–	–	–	–	–	–	
0x29 (0x49)	PLLCSR	–	–	–	–	–	PLLF	PLLE	PLOCK	31
0x28 (0x48)	OCR0B	OCR0B7	OCR0B6	OCR0B5	OCR0B4	OCR0B3	OCR0B2	OCR0B1	OCR0B0	90
0x27 (0x47)	OCR0A	OCR0A7	OCR0A6	OCR0A5	OCR0A4	OCR0A3	OCR0A2	OCR0A1	OCR0A0	90
0x26 (0x46)	TCNT0	TCNT07	TCNT06	TCNT05	TCNT04	TCNT03	TCNT02	TCNT01	TCNT00	90
0x25 (0x45)	TCCR0B	FOC0A	FOC0B	–	–	WGM02	CS02	CS01	CS00	89
0x24 (0x44)	TCCR0A	COM0A1	COM0A0	COM0B1	COM0B0	–	–	WGM01	WGM00	86
0x23 (0x43)	GTCCR	TSM	ICPSEL1	–	–	–	–	–	PSRSYNC	76
0x22 (0x42)	EEARH	–	–	–	–	–	–	EEAR9	EEAR8	20
0x21 (0x41)	EEARL	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	20
0x20 (0x40)	EEDR	EEDR7	EEDR6	EEDR5	EEDR4	EEDR3	EEDR2	EEDR1	EEDR0	20
0x1F (0x3F)	EECR	–	–	–	–	EERIE	EEMWE	EEWE	EERE	21
0x1E (0x3E)	GPIOR0	GPIOR07	GPIOR06	GPIOR05	GPIOR04	GPIOR03	GPIOR02	GPIOR01	GPIOR00	24
0x1D (0x3D)	EIMSK	–	–	–	–	INT3	INT2	INT1	INT0	71
0x1C (0x3C)	EIFR	–	–	–	–	INTF3	INTF2	INTF1	INTF0	72
0x1B (0x3B)	PCIFR	–	–	–	–	PCIF3	PCIF2	PCIF1	PCIF0	73
0x1A (0x3A)	GPIOR2	GPIOR27	GPIOR26	GPIOR25	GPIOR24	GPIOR23	GPIOR22	GPIOR21	GPIOR20	24
0x19 (0x39)	GPIOR1	GPIOR17	GPIOR16	GPIOR15	GPIOR14	GPIOR13	GPIOR12	GPIOR11	GPIOR10	24
0x18 (0x38)	Reserved	–	–	–	–	–	–	–	–	
0x17 (0x37)	Reserved	–	–	–	–	–	–	–	–	
0x16 (0x36)	TIFR1	–	–	ICF1	–	–	OCF1B	OCF1A	TOV1	115
0x15 (0x35)	TIFR0	–	–	–	–	–	OCF0B	OCF0A	TOV0	91
0x14 (0x34)	Reserved	–	–	–	–	–	–	–	–	
0x13 (0x33)	Reserved	–	–	–	–	–	–	–	–	
0x12 (0x32)	Reserved	–	–	–	–	–	–	–	–	
0x11 (0x31)	Reserved	–	–	–	–	–	–	–	–	

Notes:
1. For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.
2. I/O registers within the address range 0x00 - 0x1F 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.
3. Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.
4. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega16/32/64/M1/C1 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.
5. These registers are only available on ATmega32/64M1. For other products described in this datasheet, these locations are reserved.

34. Table of Contents