



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 "<u>Embedded -</u> <u>Microcontrollers</u>"

Details

Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	16MHz
Connectivity	SPI, UART/USART, USI
Peripherals	Brown-out Detect/Reset, LCD, POR, PWM, WDT
Number of I/O	53
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 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	64-TQFP
Supplier Device Package	64-TQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega649-16ai

Email: info@E-XFL.COM

Address: Room A, 16/F, Full Win Commercial Centre, 573 Nathan Road, Mongkok, Hong Kong



6.4 AVR Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

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

6.4.1 SREG – AVR Status Register

The AVR Status Register - SREG - is defined as:



• Bit 7 – I: Global Interrupt Enable

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

• Bit 6 – T: Bit Copy Storage

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

• Bit 5 – H: Half Carry Flag

The Half Carry Flag H indicates a Half Carry in some arithmetic operations. Half Carry Is useful in BCD arithmetic. 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 in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

• Bit 1 – Z: Zero Flag

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

14.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 14-2 shows a block diagram of the counter and its surroundings.



Figure 14-2. Counter Unit Block Diagram

Signal description (internal signals):

count	Increment or decrement TCNT0 by 1.
direction	Select between increment and decrement.
clear	Clear TCNT0 (set all bits to zero).
clk_{Tn}	Timer/Counter clock, referred to as clk_{T0} in the following.
top	Signalize that TCNT0 has reached maximum value.
bottom	Signalize that TCNT0 has reached minimum value (zero).

Depending of the mode of operation used, the counter is cleared, incremented, or decremented at each timer clock (clk_{T0}). clk_{T0} can be generated from an external or internal clock source, selected by the Clock Select bits (CS02:0). When no clock source is selected (CS02:0 = 0) the timer is stopped. However, the TCNT0 value can be accessed by the CPU, regardless of whether clk_{T0} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0A). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare output OC0A. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 97.

The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM01:0 bits. TOV0 can be used for generating a CPU interrupt.

14.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Register (OCR0A). Whenever TCNT0 equals OCR0A, the comparator signals a match. A match will set the Output Compare Flag (OCF0A) at the next timer clock cycle. If enabled (OCIE0A = 1 and Global Interrupt Flag in SREG is set), the Output Compare Flag generates an Output Compare interrupt. The OCF0A Flag is automatically cleared when the interrupt is executed. Alternatively, the OCF0A Flag can be cleared by software by writing a logical one to its I/O bit location. The





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



16.2.1 Registers

The *Timer/Counter* (TCNT1), *Output Compare Registers* (OCR1A/B), and *Input Capture Register* (ICR1) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in the section "Accessing 16-bit Registers" on page 113. The *Timer/Counter Control Registers* (TCCR1A/B) 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* (TIFR1). All interrupts are individually masked with the *Timer Interrupt Mask Register* (TIMSK1). TIFR1 and TIMSK1 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source on the T1 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_{T1}).

The double buffered Output Compare Registers (OCR1A/B) 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 (OC1A/B). See "Out-



whether clk_{T1} is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the setting of the *Waveform Generation mode* bits (WGM13:0) located in the *Timer/Counter Control Registers* A and B (TCCR1A and TCCR1B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC1x. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 123.

The Timer/Counter Overflow Flag (TOV1) is set according to the mode of operation selected by the WGM13:0 bits. TOV1 can be used for generating a CPU interrupt.

16.6 Input Capture Unit

The Timer/Counter incorporates an Input Capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP1 pin or alternatively, via the analog-comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle, and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram shown in Figure 16-3. The elements of the block diagram that are not directly a part of the Input Capture unit are gray shaded. The small "n" in register and bit names indicates the Timer/Counter number.



Figure 16-3. Input Capture Unit Block Diagram

When a change of the logic level (an event) occurs on the *Input Capture pin* (ICP1), alternatively on the *Analog Comparator output* (ACO), and this change confirms to the setting of the edge detector, a capture will be triggered. When a capture is triggered, the 16-bit value of the counter (TCNT1) is written to the *Input Capture Register* (ICR1). The *Input Capture Flag* (ICF1) is set at the same system clock as the TCNT1 value is copied into ICR1 Register. If enabled (ICIE1 = 1), the Input Capture Flag generates an Input Capture interrupt. The ICF1 Flag is automatically





The procedure for updating ICR1 differs from updating OCR1A when used for defining the TOP value. The ICR1 Register is not double buffered. This means that if ICR1 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 ICR1 value written is lower than the current value of TCNT1. 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 OCR1A Register however, is double buffered. This feature allows the OCR1A I/O location to be written anytime. When the OCR1A I/O location is written the value written will be put into the OCR1A Buffer Register. The OCR1A Compare Register will then be updated with the value in the Buffer Register at the next timer clock cycle the TCNT1 matches TOP. The update is done at the same timer clock cycle as the TCNT1 is cleared and the TOV1 Flag is set.

Using the ICR1 Register for defining TOP works well when using fixed TOP values. By using ICR1, the OCR1A Register is free to be used for generating a PWM output on OC1A. However, if the base PWM frequency is actively changed (by changing the TOP value), using the OCR1A 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 OC1x pins. Setting the COM1x1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM1x1:0 to three (see Table 16-3 on page 133). The actual OC1x value will only be visible on the port pin if the data direction for the port pin is set as output (DDR_OC1x). The PWM waveform is generated by setting (or clearing) the OC1x Register at the compare match between OCR1x and TCNT1, and clearing (or setting) the OC1x 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:

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

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

The extreme values for the OCR1x Register represents special cases when generating a PWM waveform output in the fast PWM mode. If the OCR1x is set equal to BOTTOM (0x0000) the output will be a narrow spike for each TOP+1 timer clock cycle. Setting the OCR1x equal to TOP will result in a constant high or low output (depending on the polarity of the output set by the COM1x1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC1A to toggle its logical level on each compare match (COM1A1:0 = 1). This applies only if OCR1A is used to define the TOP value (WGM13:0 = 15). The waveform generated will have a maximum frequency of $f_{OC1A} = f_{clk_l/O}/2$ when OCR1A is set to zero (0x0000). This feature is similar to the OC1A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

16.9.4 Phase Correct PWM Mode

The *phase correct Pulse Width Modulation* or phase correct PWM mode (WGM13:0 = 1, 2, 3, 10, or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency 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 (OC1x) is cleared on the compare match between TCNT1 and OCR1x while counting up, and set on the compare match while counting down. In inverting Output Compare mode, the operation is

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 16-9. 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 OCR1x and TCNT1. The OC1x Interrupt Flag will be set when a compare match occurs.





The Timer/Counter Overflow Flag (TOV1) is set at the same timer clock cycle as the OCR1x Registers are updated with the double buffer value (at BOTTOM). When either OCR1A or ICR1 is used for defining the TOP value, the OC1A or ICF1 Flag set when TCNT1 has reached TOP. The Interrupt Flags can then be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

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 OCR1x.

As Figure 16-9 shows the output generated is, in contrast to the phase correct mode, symmetrical in all periods. Since the OCR1x Registers are updated at BOTTOM, the length of the rising and the falling slopes will always be equal. This gives symmetrical output pulses and is therefore frequency correct.



When the ICR1 is used as TOP value (see description of the WGM13:0 bits located in the TCCR1A and the TCCR1B Register), the ICP1 is disconnected and consequently the Input Capture function is disabled.

• Bit 5 – Reserved Bit

This bit is reserved for future use. For ensuring compatibility with future devices, this bit must be written to zero when TCCR1B is written.

• Bit 4:3 – WGM13:2: Waveform Generation Mode

See TCCR1A Register description.

• Bit 2:0 - CS12:0: Clock Select

The three Clock Select bits select the clock source to be used by the Timer/Counter, see Figure 16-10 and Figure 16-11.

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{I/O} /1 (No prescaling)
0	1	0	clk _{I/O} /8 (From prescaler)
0	1	1	clk _{I/O} /64 (From prescaler)
1	0	0	clk _{I/O} /256 (From prescaler)
1	0	1	clk _{I/O} /1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Table 16-6. Clock Select Bit Description

If external pin modes are used for the Timer/Counter1, transitions on the T1 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

16.11.3 TCCR1C – Timer/Counter1 Control Register C



• Bit 7 – FOC1A: Force Output Compare for Unit A

• Bit 6 – FOC1B: Force Output Compare for Unit B

The FOC1A/FOC1B bits are only active when the WGM13:0 bits specifies a non-PWM mode. However, for ensuring compatibility with future devices, these bits must be set to zero when TCCR1A is written when operating in a PWM mode. When writing a logical one to the FOC1A/FOC1B bit, an immediate compare match is forced on the Waveform Generation unit. The OC1A/OC1B output is changed according to its COM1x1:0 bits setting. Note that the FOC1A/FOC1B bits are implemented as strobes. Therefore it is the value present in the COM1x1:0 bits that determine the effect of the forced compare.



PWM mode is shown in Figure 17-6. The TCNT2 value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT2 slopes represent compare matches between OCR2A and TCNT2.





The Timer/Counter Overflow Flag (TOV2) is set each time the counter reaches MAX. If the interrupt is enabled, the interrupt handler routine can be used for updating the compare value.

In fast PWM mode, the compare unit allows generation of PWM waveforms on the OC2A pin. Setting the COM2A1:0 bits to two will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM2A1:0 to three (See Table 17-4 on page 154). The actual OC2A value will only be visible on the port pin if the data direction for the port pin is set as output. The PWM waveform is generated by setting (or clearing) the OC2A Register at the compare match between OCR2A and TCNT2, and clearing (or setting) the OC2A Register at the timer clock cycle the counter is cleared (changes from MAX to BOTTOM).

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

$$f_{OCnxPWM} = \frac{f_{clk_l/O}}{N \cdot 256}$$

The N variable represents the prescale factor (1, 8, 32, 64, 128, 256, or 1024).

The extreme values for the OCR2A Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR2A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR2A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM2A1:0 bits.)

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC2A to toggle its logical level on each compare match (COM2A1:0 = 1). The waveform generated will have a maximum frequency of $f_{oc2} = f_{clk_l/O}/2$ when OCR2A is set to zero. This feature is similar to the OC2A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.







Figure 17-10 shows the setting of OCF2A in all modes except CTC mode.



Figure 17-10. Timer/Counter Timing Diagram, Setting of OCF2A, with Prescaler (f_{clk} $_{I/O}/8$)

Figure 17-11 shows the setting of OCF2A and the clearing of TCNT2 in CTC mode.

Figure 17-11. Timer/Counter Timing Diagram, Clear Timer on Compare Match mode, with Prescaler (f_{clk_l/O}/8)





18.4 Data Modes

There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figure 18-3 and Figure 18-4. Data bits are shifted out and latched in on opposite edges of the SCK signal, ensuring sufficient time for data signals to stabilize. This is clearly seen by summarizing Table 18-3 and Table 18-4, as done below:

Table 18-2. CPOL Functionality

	Leading Edge	Trailing eDge	SPI Mode
CPOL=0, CPHA=0	Sample (Rising)	Setup (Falling)	0
CPOL=0, CPHA=1	Setup (Rising)	Sample (Falling)	1
CPOL=1, CPHA=0	Sample (Falling)	Setup (Rising)	2
CPOL=1, CPHA=1	Setup (Falling)	Sample (Rising)	3







		f _{osc} = 8.0	0000MHz		f _{osc} = 11.0592MHz				f _{osc} = 14.7456MHz			
Baud Bate	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
(bps)	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	-	_	2	-7.8%	1	-7.8%	3	-7.8%
1M	_	_	0	0.0%	-	_	-	_	0	-7.8%	1	-7.8%
Max. (1)	(1) 0.5Mbps		1Mb	ps	691.2	kbps	1.3824Mbps		921.6	kbps	1.8432Mbps	

Table 19-6	Examples of LIBRRn	Settings for Commonly	v Used Oscillator Frequencies	(Continued)
		Country for Common	y obca obcinator i requeriores	(Continuou)

1. UBRR = 0, Error = 0.0%



21.2 Analog Comparator Multiplexed Input

It is possible to select any of the ADC7..0 pins to replace the negative 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) is set and the ADC is switched off (ADEN in ADCSRA is zero), MUX2..0 in ADMUX select the input pin to replace the negative input to the Analog Comparator, as shown in Table 1. If ACME is cleared or ADEN is set, AIN1 is applied to the negative input to the Analog Comparator.

ACME	ADEN	MUX20	Analog Comparator Negative Input
0	x	ХХХ	AIN1
1	1	ххх	AIN1
1	0	000	ADC0
1	0	001	ADC1
1	0	010	ADC2
1	0	011	ADC3
1	0	100	ADC4
1	0	101	ADC5
1	0	110	ADC6
1	0	111	ADC7

Table 1. Analog Comparator Multiplexed Input



• Gain Error: After adjusting for offset, the Gain Error is found as the deviation of the last transition (0x3FE to 0x3FF) compared to the ideal transition (at 1.5 LSB below maximum). Ideal value: 0 LSB

Figure 22-11. Gain Error



 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.







When EXCLK in ASSR Register is written to one, and asynchronous clock is selected, the external clock input buffer is enabled and an external clock can be input on Timer Oscillator 1 (TOSC1) pin instead of a 32kHz crystal. See "Asynchronous Operation of Timer/Counter2" on page 151 for further details.

Before entering Power-down mode, Standby mode or ADC Noise Reduction mode with synchronous LCD clock selected, the user have to disable the LCD. Refer to "Disabling the LCD" on page 237.

23.2.7 Display Blanking

When LCDBL is written to one, the LCD is blanked after completing the current frame. All segments and common pins are connected to GND, discharging the LCD. Display memory is preserved. Display blanking should be used before disabling the LCD to avoid DC voltage across segments, and a slowly fading image.

23.2.8 Port Mask

For LCD with less than 25/40 segment terminals, it is possible to mask some of the unused pins and use them as ordinary port pins instead. Refer to Table 23-3 for details. Unused common pins are automatically configured as port pins.



fer is filled one word at a time using SPM and the buffer can be filled either before the Page Erase command or between a Page Erase and a Page Write operation:

Alternative 1, fill the buffer before a Page Erase

- Fill temporary page buffer
- Perform a Page Erase
- Perform a Page Write

Alternative 2, fill the buffer after Page Erase

- Perform a Page Erase
- Fill temporary page buffer
- Perform a Page Write

If only a part of the page needs to be changed, the rest of the page must be stored (for example in the temporary page buffer) before the erase, and then be rewritten. When using alternative 1, the Boot Loader provides an effective Read-Modify-Write feature which allows the user software to first read the page, do the necessary changes, and then write back the modified data. If alternative 2 is used, it is not possible to read the old data while loading since the page is already erased. The temporary page buffer can be accessed in a random sequence. It is essential that the page address used in both the Page Erase and Page Write operation is addressing the same page. See "Simple Assembly Code Example for a Boot Loader" on page 288 for an assembly code example.

26.7.1 Performing Page Erase by SPM

To execute Page Erase, set up the address in the Z-pointer, write "X0000011" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE in the Z-register. Other bits in the Z-pointer will be ignored during this operation.

- Page Erase to the RWW section: The NRWW section can be read during the Page Erase.
- Page Erase to the NRWW section: The CPU is halted during the operation.

26.7.2 Filling the Temporary Buffer (Page Loading)

To write an instruction word, set up the address in the Z-pointer and data in R1:R0, write "00000001" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The content of PCWORD in the Z-register is used to address the data in the temporary buffer. The temporary buffer will auto-erase after a Page Write operation or by writing the RWWSRE bit in SPMCSR. It is also erased after a system reset. Note that it is not possible to write more than one time to each address without erasing the temporary buffer.

If the EEPROM is written in the middle of an SPM Page Load operation, all data loaded will be lost.

26.7.3 Performing a Page Write

To execute Page Write, set up the address in the Z-pointer, write "X0000101" to SPMCSR and execute SPM within four clock cycles after writing SPMCSR. The data in R1 and R0 is ignored. The page address must be written to PCPAGE. Other bits in the Z-pointer must be written to zero during this operation.

- Page Write to the RWW section: The NRWW section can be read during the Page Write.
- Page Write to the NRWW section: The CPU is halted during the operation.



26.7.12 Simple Assembly Code Example for a Boot Loader

```
;-the routine writes one page of data from RAM to Flash
 ; the first data location in RAM is pointed to by the Y pointer
 ; the first data location in Flash is pointed to by the Z-pointer
 ;-error handling is not included
 ;-the routine must be placed inside the Boot space
 ; (at least the Do_spm sub routine). Only code inside NRWW section can
 ; be read during Self-Programming (Page Erase and Page Write).
 ;-registers used: r0, r1, temp1 (r16), temp2 (r17), looplo (r24),
 ; loophi (r25), spmcrval (r20)
 ; storing and restoring of registers is not included in the routine
 ; register usage can be optimized at the expense of code size
 ;-It is assumed that either the interrupt table is moved to the Boot
 ; loader section or that the interrupts are disabled.
.equ PAGESIZEB = PAGESIZE*2 ; PAGESIZEB is page size in BYTES, not words
.org SMALLBOOTSTART
Write_page:
 ; Page Erase
 ldi spmcrval, (1<<PGERS) | (1<<SPMEN)</pre>
 call Do_spm
 ; re-enable the RWW section
 ldi spmcrval, (1<<RWWSRE) | (1<<SPMEN)
 call Do_spm
 ; transfer data from RAM to Flash page buffer
 ldi looplo, low(PAGESIZEB) ;init loop variable
 ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
Wrloop:
 ld r0, Y+
 ld r1, Y+
 ldi spmcrval, (1<<SPMEN)
 call Do_spm
 adiw ZH:ZL, 2
 sbiw loophi:looplo, 2
                               ;use subi for PAGESIZEB<=256
 brne Wrloop
 ; execute Page Write
 subi ZL, low(PAGESIZEB)
                               ;restore pointer
 sbci ZH, high(PAGESIZEB)
                               ;not required for PAGESIZEB<=256
 ldi spmcrval, (1<<PGWRT) | (1<<SPMEN)
 call Do_spm
 ; re-enable the RWW section
 ldi spmcrval, (1<<RWWSRE) | (1<<SPMEN)
 call Do_spm
 ; read back and check, optional
 ldi looplo, low(PAGESIZEB) ;init loop variable
 ldi loophi, high(PAGESIZEB) ;not required for PAGESIZEB<=256
 subi YL, low(PAGESIZEB)
                               ;restore pointer
 sbci YH, high(PAGESIZEB)
Rdloop:
 lpm r0, Z+
 ld r1, Y+
 cpse r0, r1
```

30. Register Summary

Note: Registers with bold type only available in ATmega3290/6490.

		D -	B !! A	5	54	5	5	5.4	5	2
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	LCDDR19	SEG339	SEG338	SEG337	SEG336	SEG335	SEG334	SEG333	SEG332	244
(0xFE)	LCDDR18	SEG331	SEG330	SEG329	SEG328	SEG327	SEG326	SEG325	SEG324	244
(0xFD)	LCDDR17	SEG323	SEG322	SEG321	SEG320	SEG319	SEG318	SEG317	SEG316	244
(0xFC)	LCDDR16	SEG315	SEG314	SEG313	SEG312	SEG311	SEG310	SEG309	SEG308	244
(0xFB)	LCDDR15	SEG307	SEG306	SEG305	SEG304	SEG303	SEG302	SEG301	SEG300	244
(0xFA)	LCDDR14	SEG239	SEG238	SEG237	SEG236	SEG235	SEG234	SEG233	SEG232	244
(0xF9)	LCDDR13	SEG231	SEG230	SEG229	SEG228	SEG227	SEG226	SEG225	SEG224	244
(0xF8)	LCDDR12	SEG223	SEG222	SEG221	SEG220	SEG219	SEG218	SEG217	SEG216	244
(0xF7)	LCDDR11	SEG215	SEG214	SEG213	SEG212	SEG211	SEG210	SEG209	SEG208	244
(0xF6)	LCDDR10	SEG207	SEG206	SEG205	SEG204	SEG203	SEG202	SEG201	SEG200	244
(0xF5)	LCDDR09	SEG139	SEG138	SEG137	SEG136	SEG135	SEG134	SEG133	SEG132	244
(0xF4)	LCDDR08	SEG131	SEG130	SEG129	SEG128	SEG127	SEG126	SEG125	SEG124	244
(0xE3)	LCDDR07	SEG123	SEG122	SEG121	SEG120	SEG119	SEG118	SEG117	SEG116	244
(0xF2)	LCDDR06	SEG115	SEG114	SEG113	SEG112	SEG111	SEG110	SEG109	SEG108	244
(0xE1)	LCDDR05	SEG107	SEG106	SEG105	SEG104	SEG103	SEG102	SEG101	SEG100	244
(0xE0)	LCDDR04	SEG039	SEG038	SEG037	SEG036	SEG035	SEG034	SEG033	SEG032	244
(0xEE)	LCDDB03	SEG031	SEG030	SEG029	SEG028	SEG027	SEG026	SEG025	SEG024	244
	LCDDB02	SEG023	SEG022	SEG021	SEG020	SEG019	SEG018	SEG017	SEG016	244
(UXEE)	LCDDB01	SEG015	SEG014	SEG013	SEG012	SEG011	SEG010	SEG009	SEG008	244
(0xED)		SEG007	SEG006	SEG005	SEG004	SEG003	SEG002	SEG001	SEG000	244
(UXEC)	Beconvod	32007	320000	320003	320004	320003	320002	320001	320000	244
(UXEB)	Reserved	-	-	-	-	-	-	-	-	
(0xEA)	Reserved	-	-	-	-	-	-	-	-	
(0xE9)	Reserved	-	-	-	-	-	-	-	-	
(0xE8)	Reserved	-	-	-	-	-	-	-	-	0.40
(0xE7)	LCDCCR	LCDDC2	LCDDC1	LCDDC0	-	LCDCC3	LCDCC2	LCDCC1	LCDCCO	243
(0xE6)	LCDFRR	-	LCDPS2	LCDPS1	LCDPS0	-	LCDCD2	LCDCD1	LCDCD0	241
(0xE5)	LCDCRB	LCDCS	LCD2B	LCDMUX1	LCDMUX0	LCDPM3	LCDPM2	LCDPM1	LCDPM0	239
(0xE4)	LCDCRA	LCDEN	LCDAB	-	LCDIF	LCDIE	-	-	LCDBL	239
(0xE3)	Reserved	-	-	-	-	-	-	-	-	
(0xE2)	Reserved	-	-	-	-	-	-	-	-	
(0xE1)	Reserved	-	-	-	-	-	-	-	-	
(0xE0)	Reserved	-	-	-	-	-	-	-	-	
(0xDF)	Reserved	-	-	-	-	-	-	-	-	
(0xDE)	Reserved	-	-	-	-	-	-	-	-	
(0xDD)	PORTJ	-	PORTJ6	PORTJ5	PORTJ4	PORTJ3	PORTJ2	PORTJ1	PORTJ0	90
(0xDC)	DDRJ	-	DDJ6	DDJ5	DDJ4	DDJ3	DDJ2	DDJ1	DDJ0	90
(0xDB)	PINJ	-	PINJ6	PINJ5	PINJ4	PINJ3	PINJ2	PINJ1	PINJ0	90
(0xDA)	PORTH	PORTH7	PORTH6	PORTH5	PORTH4	PORTH3	PORTH2	PORTH1	PORTH0	89
(0xD9)	DDRH	DDH7	DDH6	DDH5	DDH4	DDH3	DDH2	DDH1	DDH0	90
(0xD8)	PINH	PINH7	PINH6	PINH5	PINH4	PINH3	PINH2	PINH1	PINH0	90
(0xD7)	Reserved	-	-	-	-	-	-	-	-	
(0xD6)	Reserved	-	-	-	-	-	-	-	-	
(0xD5)	Reserved	-	-	-	-	-	-	-	-	
(0xD4)	Reserved	-	-	-	-	-	-	-	-	
(0xD3)	Reserved	-	-	-	-	-	-	-	-	
(0xD2)	Reserved	-	-	-	-	-	-	-	-	
(0xD1)	Reserved	-	-	-	-	-	-	-	-	
(0xD0)	Reserved	-	-	-	-	-	-	-	-	
(0xCF)	Reserved	-	-	-	-	-	-	-	-	
(0xCE)	Reserved	-	-	-	-	-	-	-	-	
(0xCD)	Reserved	-	-	-	-	-	-	-	-	
(0xCC)	Reserved	-	-	-	-	-	-	-	-	
(0xCB)	Reserved	-	-	-	-	-	-	-	-	
	Reserved	-	-	-	-	-	-	-	-	
	Reserved	-	-	-	-	-	-	-	-	
(0xC9)	Reserved	_	-	-	-	-	-	-	-	
(0x08)	Reserved								_	
(UXC7)	LIDBU				LISARTOD	ata Begister				190
(UXC6)					COANTO D			ate Begister Lich		104
(UXC5)						Poto Register Law	JUNATU Daud H	are negister migh		194
(0xC4)	UDKKUL	USART0 Baud Rate Register Low 1								194



	13.4	Register Description	87
14	8-bit Tir	ner/Counter0 with PW M	
	14.1	Features	91
	14.2	Overview	91
	14.3	Timer/Counter Clock Sources	92
	14.4	Counter Unit	93
	14.5	Output Compare Unit	93
	14.6	Compare Match Output Unit	95
	14.7	Modes of Operation	97
	14.8	Timer/Counter Timing Diagrams	101
	14.9	Register Description	103
15	Timer/C	Counter0 and Time r/Counter1 Prescalers	107
	15.1	Register Description	
16	16-bit T	imer/Counter1	110
	16.1	Features	110
	16.2	Overview	110
	16.3	Accessing 16-bit Registers	113
	16.4	Timer/Counter Clock Sources	116
	16.5	Counter Unit	116
	16.6	Input Capture Unit	117
	16.7	Output Compare Units	119
	16.8	Compare Match Output Unit	122
	16.9	Modes of Operation	123
	16.10	Timer/Counter Timing Diagrams	
	16.11	Register Description	132
17	8-bit Tir	ner/Counter2 with PW M and Asynchronous Operation	139
	17.1	Features	139
	17.2	Overview	139
	17.3	Timer/Counter Clock Sources	140
	17.4	Counter Unit	140
	17.5	Output Compare Unit	141
	17.6	Compare Match Output Unit	144
	17.7	Modes of Operation	145
	17.8	Timer/Counter Timing Diagrams	149
	17.9	Asynchronous Operation of Timer/Counter2	151

