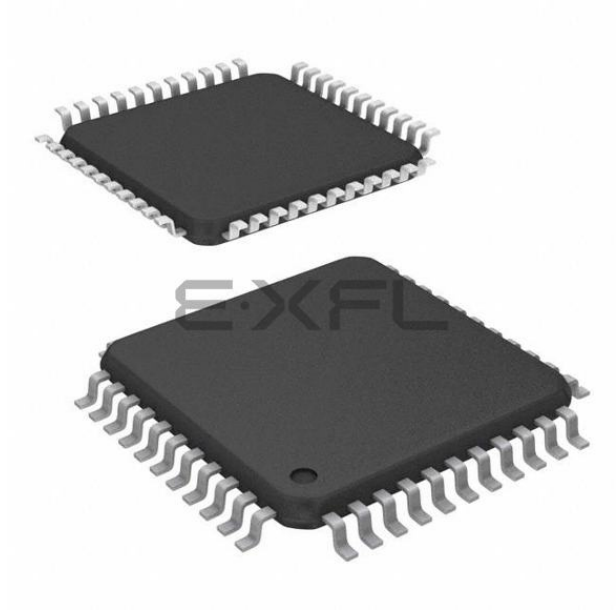


Welcome to E-XFL.COM

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	8/16-Bit
Speed	32MHz
Connectivity	I²C, IrDA, SPI, UART/USART
Peripherals	Brown-out Detect/Reset, POR, PWM, WDT
Number of I/O	34
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)	1.6V ~ 3.6V
Data Converters	A/D 12x12b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 105°C (TA)
Mounting Type	Surface Mount
Package / Case	44-TQFP
Supplier Device Package	44-TQFP (10x10)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atxmega64d4-an

3.5 Program Flow

After reset, the CPU starts to execute instructions from the lowest address in the flash program memory '0.' The program counter (PC) addresses the next instruction to be fetched.

Program flow is provided by conditional and unconditional jump and call instructions capable of addressing the whole address space directly. Most AVR instructions use a 16-bit word format, while a limited number use a 32-bit format.

During interrupts and subroutine calls, the return address PC is stored on the stack. The stack is allocated in the general data SRAM, and consequently the stack size is only limited by the total SRAM size and the usage of the SRAM. After reset, the stack pointer (SP) points to the highest address in the internal SRAM. The SP is read/write accessible in the I/O memory space, enabling easy implementation of multiple stacks or stack areas. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR CPU.

3.6 Instruction Execution Timing

The AVR CPU is clocked by the CPU clock, clk_{CPU}. No internal clock division is used. Figure 3-2 shows the parallel instruction fetches and instruction executions enabled by the Harvard architecture and the fast-access register file concept. This is the basic pipelining concept used to obtain up to 1MIPS/MHz performance with high power efficiency.

Figure 3-2. The Parallel Instruction Fetches and Instruction Executions

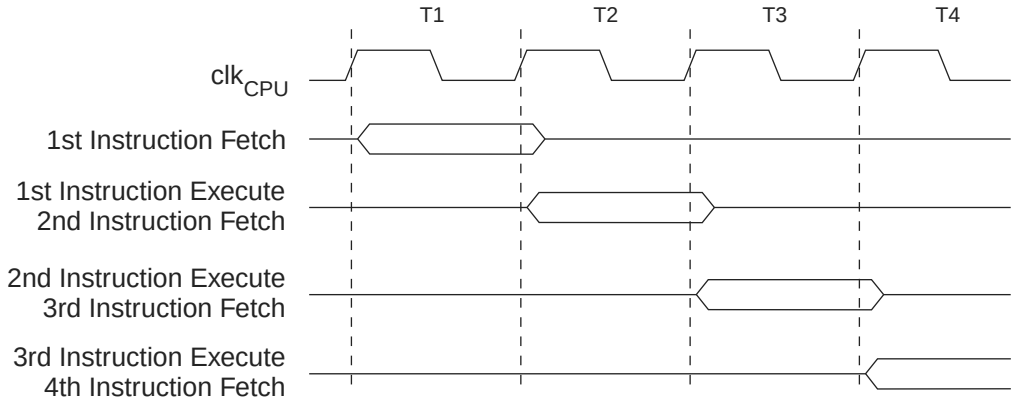
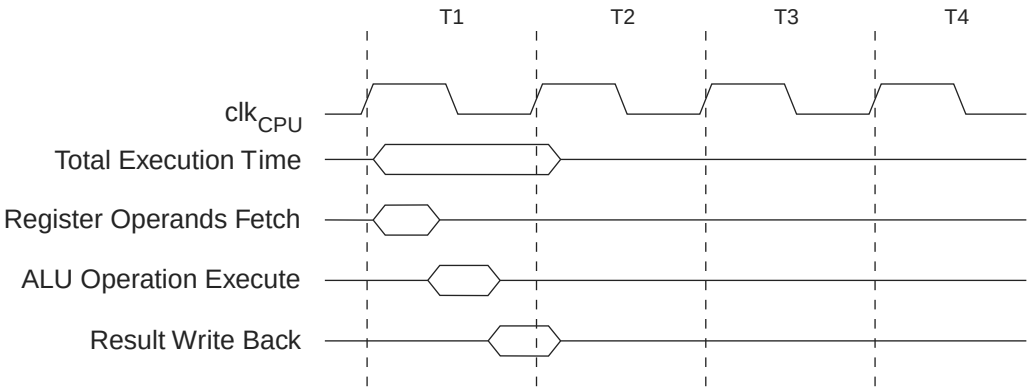


Figure 3-3 shows the internal timing concept for the register file. In a single clock cycle, an ALU operation using two register operands is executed and the result is stored back to the destination register.

Figure 3-3. Single Cycle ALU Operation



- **Bit 4:2 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to one when this register is written.

- **Bit 1:0 – BODPD[1:0]: BOD Operation in Power-down Mode**

These fuse bits set the BOD operation mode in all sleep modes except idle mode.

For details on the BOD and BOD operation modes, refer to “Brownout Detection” on page 83.

Table 4-3. BOD Operation Modes in Sleep Modes

BODPD[1:0]	Description
00	Reserved
01	BOD enabled in sampled mode
10	BOD enabled continuously
11	BOD disabled

4.13.3 FUSEBYTE4 – Fuse Byte 4

Bit	7	6	5	4	3	2	1	0
+0x04	–	–	–	RSTDISBL	STARTUPTIME[1:0]		WDLOCK	–
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	1	1	1	1	1	1	1	1

- **Bit 7:5 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to one when this register is written.

- **Bit: 4 – RSTDISBL: External Reset Disable**

This fuse can be programmed to disable the external reset pin functionality. When this is done, pulling the reset pin low will not cause an external reset. A reset is required before this bit will be read correctly after it is changed.

- **Bit 3:2 – STARTUPTIME[1:0]: Start-up time**

These fuse bits can be used to set at a programmable timeout period from when all reset sources are released until the internal reset is released from the delay counter. A reset is required before these bits will be read correctly after they are changed.

The delay is timed from the 1kHz output of the ULP oscillator. Refer to “Reset Sequence” on page 82 for details.

Table 4-4. Start-up Time

STARTUPTIME[1:0	1kHz ULP oscillator cycles
00	64
01	4
10	Reserved
11	0

- **Bit 1 – WDLOCK: Watchdog Timer Lock**

The WDLOCK fuse can be programmed to lock the watchdog timer configuration. When this fuse is programmed, the watchdog timer configuration cannot be changed, and the ENABLE bit in the watchdog CTRL register is automatically set at reset and cannot be cleared from the application software. The WEN bit in the watchdog WINCTRL register is not set

5.8.2 CHnCTRL – Event Channel n Control Register

Bit	7	6	5	4	3	2	1	0
	–	QDIRM[1:0](1)		QDIEN(1)	QDEN(1)	DIGFILT[2:0]		
	–	–	–	–	–	DIGFILT[2:0]		
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R
Initial Value	0	0	0	0	0	0	0	0

Note: 1. Only available for CH0CTRL and CH2CTRL. These bits are reserved in CH1CTRL and CH3CTRL.

- **Bit 7 – Reserved**

This bit is reserved and will always be read as zero. For compatibility with future devices, always write this bit to zero when this register is written.

- **Bit 6:5 – QDIRM[1:0]: Quadrature Decode Index Recognition Mode**

These bits determine the quadrature state for the QDPH0 and QDPH90 signals, where a valid index signal is recognized and the counter index data event is given according to Table 5-5. These bits should only be set when a quadrature encoder with a connected index signal is used.These bits are available only for CH0CTRL and CH2CTRL.

Table 5-5. QDIRM Bit Settings

QDIRM[1:0]		Index recognition state
0	0	{QDPH0, QDPH90} = 0b00
0	1	{QDPH0, QDPH90} = 0b01
1	0	{QDPH0, QDPH90} = 0b10
1	1	{QDPH0, QDPH90} = 0b11

- **Bit 4 – QDIEN: Quadrature Decode Index Enable**

When this bit is set, the event channel will be used as a QDEC index source, and the index data event will be enabled.

This bit is available only for CH0CTRL and CH2CTRL.

- **Bit 3 – QDEN: Quadrature Decode Enable**

Setting this bit enables QDEC operation.

This bit is available only for CH0CTRL and CH2CTRL.

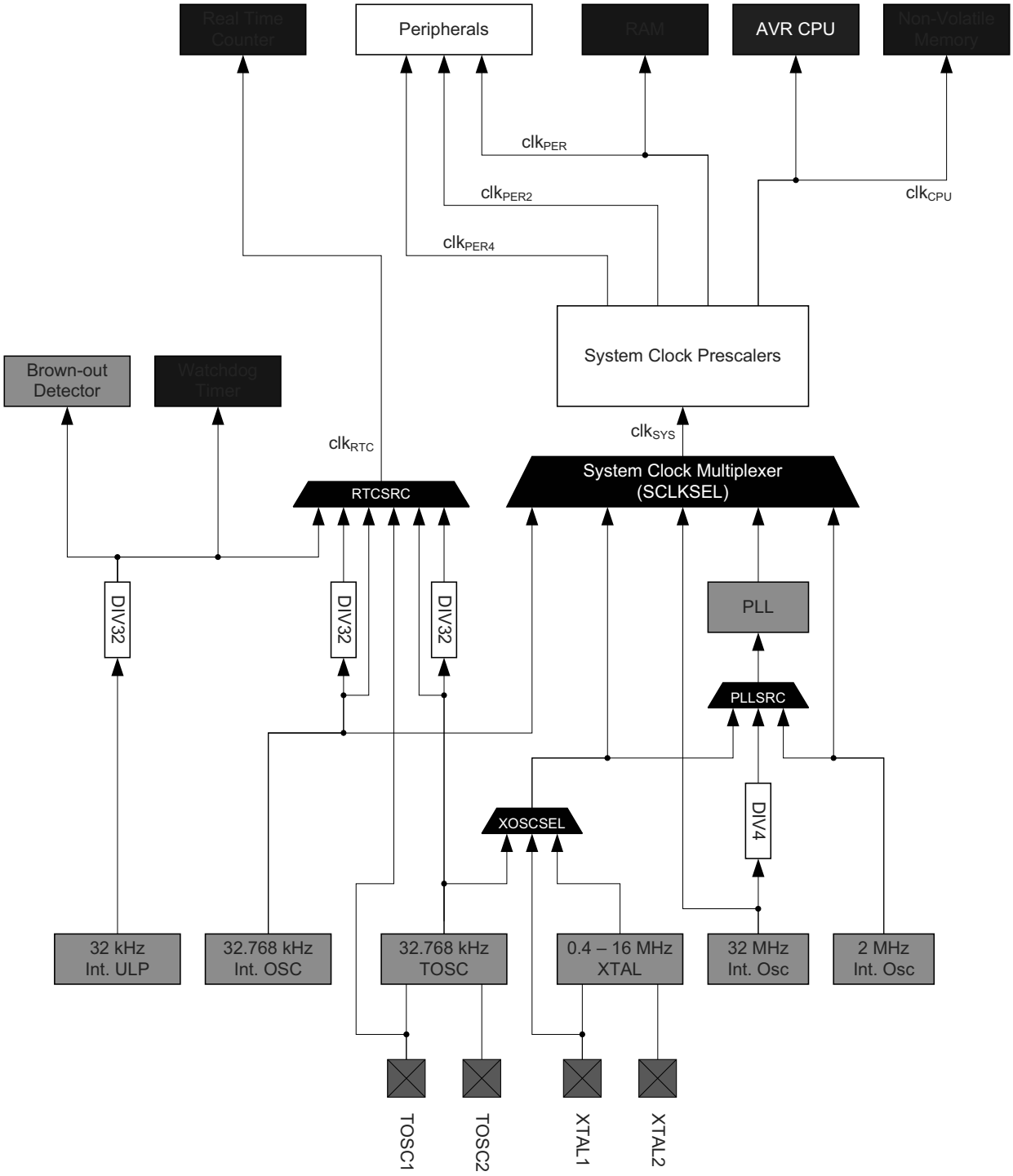
- **Bit 2:0 – DIGFILT[2:0]: Digital Filter Coefficient**

These bits define the length of digital filtering used, according to Table 5-6. Events will be passed through to the event channel only when the event source has been active and sampled with the same level for the number of peripheral clock cycles defined by DIGFILT.

Table 5-6. Digital Filter Coefficient Values

DIGFILT[2:0]	Group configuration	Description
000	1SAMPLE	One sample
001	2SAMPLES	Two samples
010	3SAMPLES	Three samples
011	4SAMPLES	Four samples
100	5SAMPLES	Five samples

Figure 6-1. The Clock System, Clock Sources, and Clock Distribution



6.9 Register Description – Clock

6.9.1 CTRL – Control Register

Bit	7	6	5	4	3	2	1	0
+0x00	–	–	–	–	–	SCLKSEL[2:0]		
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:3 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 2:0 – SCLKSEL[2:0]: System Clock Selection**

These bits are used to select the source for the system clock. See Table 6-1 for the different selections. Changing the system clock source will take two clock cycles on the old clock source and two more clock cycles on the new clock source. These bits are protected by the configuration change protection mechanism. For details, refer to “Configuration Change Protection” on page 13.

SCLKSEL cannot be changed if the new clock source is not stable. The old clock can not be disabled until the clock switching is completed.

Table 6-1. System Clock Selection

SCLKSEL[2:0]	Group configuration	Description
000	RC2MHZ	2MHz internal oscillator
001	RC32MHZ	32MHz internal oscillator
010	RC32KHZ	32.768kHz internal oscillator
011	XOSC	External oscillator or clock
100	PLL	Phase locked loop
101	–	Reserved
110	–	Reserved
111	–	Reserved

6.9.2 PSCTRL – Prescaler Register

This register is protected by the configuration change protection mechanism. For details, refer to “Configuration Change Protection” on page 13.

Bit	7	6	5	4	3	2	1	0
+0x01	–	PSADIV[4:0]					PSBCDIV	
Read/Write	R	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 – Reserved**

This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.

10.8.3 CTRL – Control Register

Bit	7	6	5	4	3	2	1	0
+0x02	**RREN**	**IVSEL**	**–**	**–**	**–**	**HILVLEN**	**MEDLVLEN**	**LOLVLEN**
Read/Write	R/W	R/W	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7 – RREN: Round-robin Scheduling Enable**

When the RREN bit is set, the round-robin scheduling scheme is enabled for low-level interrupts. When this bit is cleared, the priority is static according to interrupt vector address, where the lowest address has the highest priority.

- **Bit 6 – IVSEL: Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the interrupt vectors are placed at the start of the application section in flash. When this bit is set (one), the interrupt vectors are placed in the beginning of the boot section of the flash. Refer to the device datasheet for the absolute address.

This bit is protected by the configuration change protection mechanism. Refer to “Configuration Change Protection” on page 13 for details.

- **Bit 5:3 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 2 – HILVLEN: High-level Interrupt Enable** (1)

When this bit is set, all high-level interrupts are enabled. If this bit is cleared, high-level interrupt requests will be ignored.

- **Bit 1 – MEDLVLEN: Medium-level Interrupt Enable**(1)

When this bit is set, all medium-level interrupts are enabled. If this bit is cleared, medium-level interrupt requests will be ignored.

- **Bit 0 – LOLVLEN: Low-level Interrupt Enable**(1)

When this bit is set, all low-level interrupts are enabled. If this bit is cleared, low-level interrupt requests will be ignored.

Note: 1. Ignoring interrupts will be effective one cycle after the bit is cleared.

10.9 Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
+0x00	STATUS	NMIEX	–	–	–	–	HILVLEX	MEDLVLEX	LOLVLEX	99
+0x01	INTPRI	INTPRI[7:0]								99
+0x02	CTRL	RREN	IVSEL	–	–	–	HILVLEN	MEDLVLEN	LOLVLEN	100

11.12.9 IN – Data Input Value Register

Bit	7	6	5	4	3	2	1	0
+0x08	IN[7:0]							
Read/Write	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:0 – IN[7:0]: Data Input Value**

This register shows the value present on the pins if the digital input driver is enabled. INn shows the value of pin n of the port. The input is not sampled and cannot be read if the digital input buffers are disabled.

11.12.10INTCTRL – Interrupt Control Register

Bit	7	6	5	4	3	2	1	0
+0x09	–	–	–	–	INT1LVL[1:0]		INT0LVL[1:0]	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:4 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:2/1:0 – INTnLVL[1:0]: Interrupt n Level**

These bits enable port interrupt n and select the interrupt level as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94.

11.12.11INT0MASK – Interrupt 0 Mask Register

Bit	7	6	5	4	3	2	1	0
+0x0A	INT0MSK[7:0]							
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 7:0 – INT0MSK[7:0]: Interrupt 0 Mask bits**

These bits are used to mask which pins can be used as sources for port interrupt 0. If INT0MASKn is written to one, pin n is used as source for port interrupt 0.The input sense configuration for each pin is decided by the PINnCTRL registers.

11.12.12INT1MASK – Interrupt 1 Mask Register

Bit	7	6	5	4	3	2	1	0
+0x0B	INT1MSK[7:0]							
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 7:0 – INT1MASK[7:0]: Interrupt 1 Mask bits**

These bits are used to mask which pins can be used as sources for port interrupt 1. If INT1MASKn is written to one, pin n is used as source for port interrupt 1.The input sense configuration for each pin is decided by the PINnCTRL registers.

11.13 Register Descriptions – Port Configuration

11.13.1 MPCMASK – Multi-pin Configuration Mask Register

Bit	7	6	5	4	3	2	1	0
+0x00	MPCMASK[7:0]							
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 7:0 – MPCMASK[7:0]: Multi-pin Configuration Mask**

The MPCMASK register enables configuration of several pins of a port at the same time. Writing a one to bit n makes pin n part of the multi-pin configuration. When one or more bits in the MPCMASK register is set, writing any of the PINnCTRL registers will update only the PINnCTRL registers matching the mask in the MPCMASK register for that port. The MPCMASK register is automatically cleared after any PINnCTRL register is written.

11.13.2 VPCTRLA 116– Virtual Port-map Control Register A

Bit	7	6	5	4	3	2	1	0
+0x02	VP1MAP[3:0]				VP0MAP[3:0]			
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 7:4 – VP1MAP: Virtual Port 1 Mapping**

These bits decide which ports should be mapped to Virtual Port 1. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 11-7 on page 117 for configuration.

- **Bit 3:0 – VP0MAP: Virtual Port 0 Mapping**

These bits decide which ports should be mapped to Virtual Port 0. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 11-7 on page 117 for configuration.

11.13.3 VPCTRLB – Virtual Port-map Control Register B

Bit	7	6	5	4	3	2	1	0
+0x03	VP3MAP[3:0]				VP2MAP[3:0]			
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

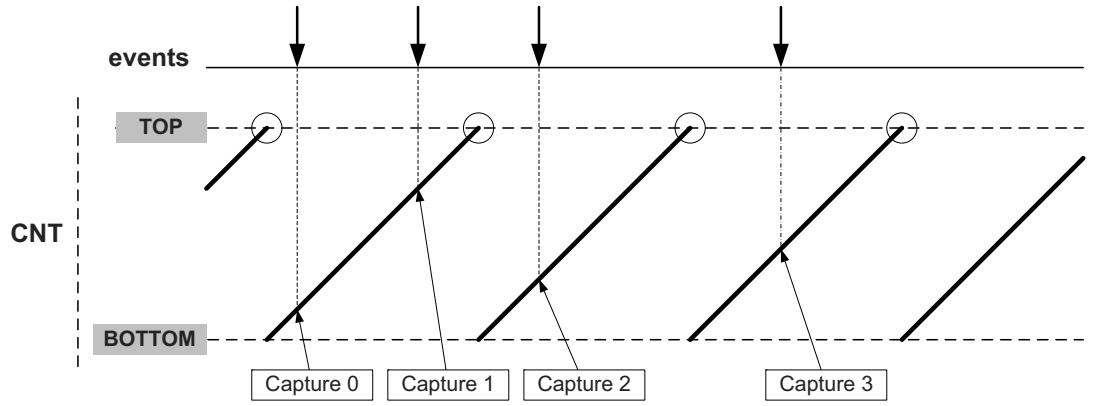
- **Bit 7:4 – VP3MAP: Virtual Port 3 Mapping**

These bits decide which ports should be mapped to Virtual Port 3. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 11-7 on page 117 for configuration.

- **Bit 3:0 – VP2MAP: Virtual Port 2 Mapping**

These bits decide which ports should be mapped to Virtual Port 2. The registers DIR, OUT, IN, and INTFLAGS will be mapped. Accessing the virtual port registers is equal to accessing the actual port registers. See Table 11-7 on page 117 for configuration.

Figure 12-11.Input Capture Timing



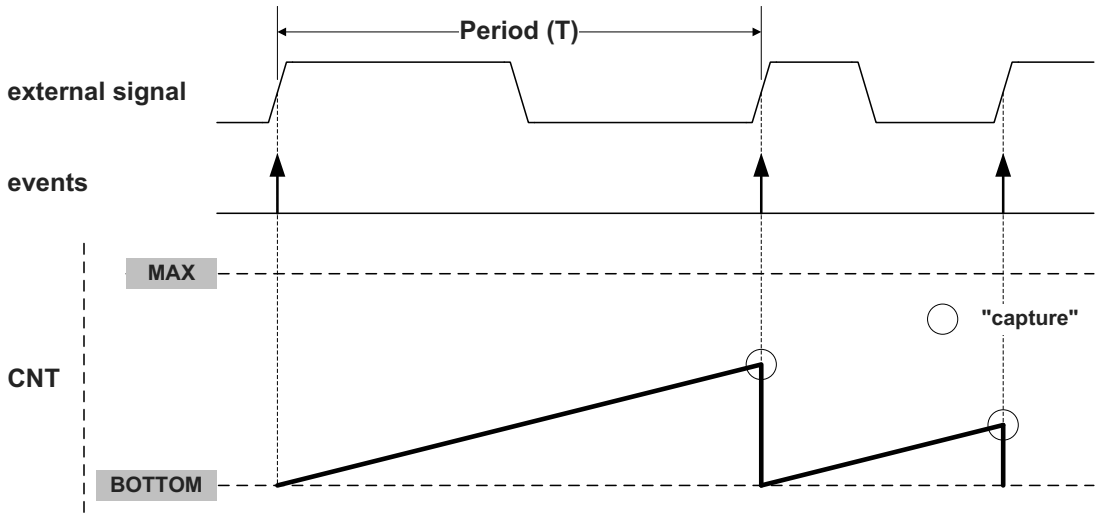
12.7.2 Frequency Capture

Selecting the frequency capture event action makes the enabled capture channel perform an input capture and restart on positive edge events. This enables the timer/counter to measure the period or frequency of a signal directly. The capture result will be the time (T) from the previous timer/counter restart until the event occurred. This can be used to calculate the frequency (f) of the signal:

$$f = \frac{1}{T}$$

Figure 12-12 shows an example where the period of an external signal is measured twice.

Figure 12-12.Frequency Capture of an External Signal



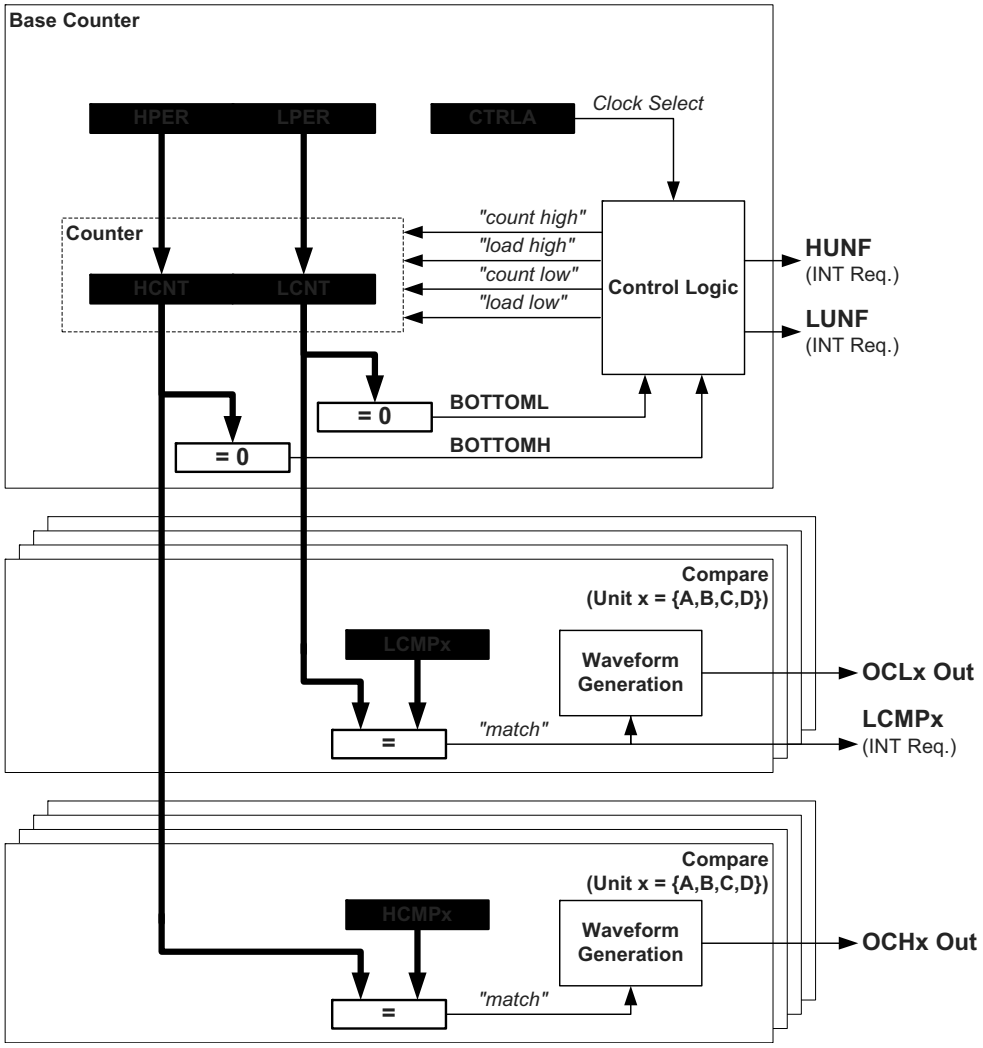
Since all capture channels use the same counter (CNT), only one capture channel must be enabled at a time. If two capture channels are used with different sources, the counter will be restarted on positive edge events from both input sources, and the result will have no meaning.

12.7.3 Pulse Width Capture

Selecting the pulse width measure event action makes the enabled compare channel perform the input capture action on falling edge events and the restart action on rising edge events. The counter will then restart on positive edge events, and the input capture will be performed on the negative edge event. The event source must be an I/O pin, and the sense configuration for the pin must be set to generate an event on both edges. Figure 12-13 on page 131 shows and example where the pulse width is measured twice for an external signal.

13.3 Block Diagram

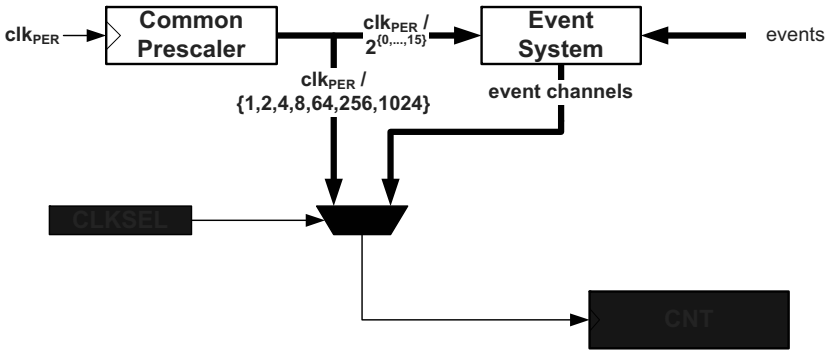
Figure 13-1. Block Diagram of the 16-bit Timer/counter 0 with Split Mode



13.4 Clock Sources

The timer/counter can be clocked from the peripheral clock (clk_{PER}) and from the event system. Figure 13-2 shows the clock and event selection.

Figure 13-2. Clock Selection



13.9.6 INTCTRLB – Interrupt Enable Register B

Bit	7	6	5	4	3	2	1	0
+0x07	LCMPDINTLVL[1:0]		LCMPCINTLVL[1:0]		LCMPBINTLVL[1:0]		LCMPAINTLVL[1:0]	
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 7:0 – LCMPxINTLVL[1:0]: Low-byte Compare x Interrupt Level**

These bits enable the low-byte timer compare interrupt and select the interrupt level, as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94. The enabled interrupt will be triggered when LCMPxIF in the INTFLAGS register is set.

13.9.7 CTRLF – Control Register F

Bit	7	6	5	4	3	2	1	0
+0x08	–	–	–	–	CMD[1:0]		CMDEN[1:0]	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:4 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:2 – CMD[1:0]: Timer/Counter Command**

These command bits are used for software control of timer/counter update, restart, and reset. The command bits are always read as zero. The CMD bits must be used together with CMDEN.

Table 13-3. Command Selections

CMD	Group configuration	Description
00	NONE	None
01	—	Reserved
10	RESTART	Force restart
11	RESET	Force hard reset (ignored if T/C is not in OFF state)

- **Bit 1:0 – CMDEN[1:0]: Command Enable**

These bits are used to indicate for which timer/counter the command (CMD) is valid.

Table 13-4. Command Enable Selections

CMDEN	Group configuration	Description
00	–	Reserved
01	LOW	Command valid for low-byte T/C
10	HIGH	Command valid for high-byte T/C
11	BOTH	Command valid for both low-byte and high-byte T/C

17.10 Register Description – TWI Slave

17.10.1 CTRLA – Control Register A

Bit	7	6	5	4	3	2	1	0
+0x00	INTLVL[1:0]		DIEN	APIEN	ENABLE	PIEN	PMEN	SMEN
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 7:6 – INTLVL[1:0]: Interrupt Level**

These bits select the interrupt level for the TWI master interrupt, as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94.

- **Bit 5 – DIEN: Data Interrupt Enable**

Setting the data interrupt enable (DIEN) bit enables the data interrupt when the data interrupt flag (DIF) in the STATUS register is set. The INTLVL bits must be nonzero for the interrupt to be generated.

- **Bit 4 – APIEN: Address/Stop Interrupt Enable**

Setting the address/stop interrupt enable (APIEN) bit enables the address/stop interrupt when the address/stop interrupt flag (APIF) in the STATUS register is set. The INTLVL bits must be nonzero for interrupt to be generated.

- **Bit 3 – ENABLE: Enable TWI Slave**

Setting this bit enables the TWI slave.

- **Bit 2 – PIEN: Stop Interrupt Enable**

Setting the this bit will cause APIF in the STATUS register to be set when a STOP condition is detected.

- **Bit 1 – PMEN: Promiscuous Mode Enable**

By setting the this bit, the slave address match logic responds to all received addresses. If this bit is cleared, the address match logic uses the ADDR register to determine which address to recognize as its own address.

- **Bit 0 – SMEN: Smart Mode Enable**

This bit enables smart mode. When Smart mode is enabled, the acknowledge action, as set by the ACKACT bit in the CTRLB register, is sent immediately after reading the DATA register.

17.10.2 CTRLB – Control Register B

Bit	7	6	5	4	3	2	1	0
+0x01	–	–	–	–	–	ACKACT	CMD[1:0]	
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:3 – Reserved**

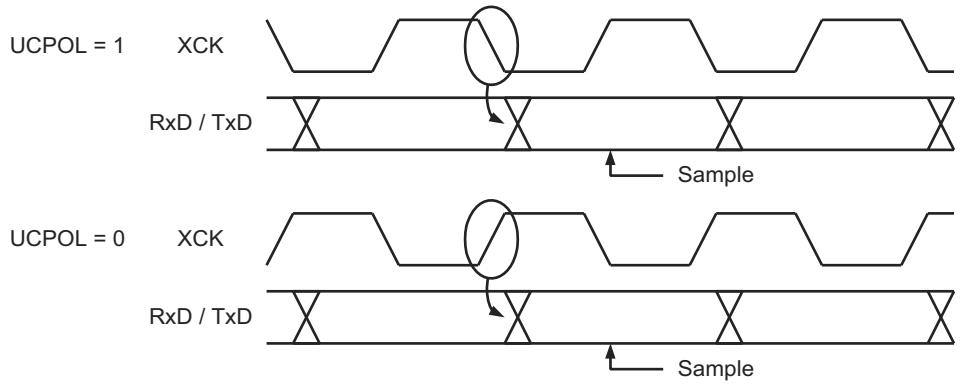
These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 2 – ACKACT: Acknowledge Action**

This bit defines the slave's acknowledge behavior after an address or data byte is received from the master. The acknowledge action is executed when a command is written to the CMD bits. If the SMEN bit in the CTRLA register is set, the acknowledge action is performed when the DATA register is read.

Table 17-7 on page 192 lists the acknowledge actions.

Figure 19-3. Synchronous Mode XCK Timing



Using the inverted I/O (INVEN) setting for the corresponding XCK port pin, the XCK clock edges used for data sampling and data change can be selected. If inverted I/O is disabled (INVEN=0), data will be changed at the rising XCK clock edge and sampled at the falling XCK clock edge. If inverted I/O is enabled (INVEN=1), data will be changed at the falling XCK clock edge and sampled at the rising XCK clock edge. For more details, see “I/O Ports” on page 101.

19.3.5 Master SPI Mode Clock Generation

For master SPI mode operation, only internal clock generation is supported. This is identical to the USART synchronous master mode, and the baud rate or BSEL setting is calculated using the same equations (see Table 19-1 on page 204).

There are four combinations of the SPI clock (SCK) phase and polarity with respect to the serial data, and these are determined by the clock phase (UCPHA) control bit and the inverted I/O pin (INVEN) settings. The data transfer timing diagrams are shown in Figure 19-4 on page 207. Data bits are shifted out and latched in on opposite edges of the XCK signal, ensuring sufficient time for data signals to stabilize. The UCPHA and INVEN settings are summarized in Table 19-2. Changing the setting of any of these bits during transmission will corrupt both the receiver and transmitter

Table 19-2. INVEN and UCPHA Functionality

SPI mode	INVEN	UCPHA	Leading edge	Trailing edge
0	0	0	Rising, sample	Falling, setup
1	0	1	Rising, setup	Falling, sample
2	1	0	Falling, sample	Rising, setup
3	1	1	Falling, setup	Rising, sample

The leading edge is the first clock edge of a clock cycle. The trailing edge is the last clock edge of a clock cycle.

- **Bit 4 – FERR: Frame Error**

The FERR flag indicates the state of the first stop bit of the next readable frame stored in the receive buffer. The bit is set if the received character had a frame error, i.e., the first stop bit was zero, and cleared when the stop bit of the received data is one. This bit is valid until the receive buffer (DATA) is read. FERR is not affected by setting the number of stop bits used, as it always uses only the first stop bit. Always write this bit location to zero when writing the STATUS register.

This flag is not used in master SPI mode operation.

- **Bit 3 – BUFOVF: Buffer Overflow**

This flag indicates data loss due to a receiver buffer full condition. This flag is set if a buffer overflow condition is detected. A buffer overflow occurs when the receive buffer is full (two characters) with a new character waiting in the receive shift register and a new start bit is detected. This flag is valid until the receive buffer (DATA) is read. Always write this bit location to zero when writing the STATUS register.

This flag is not used in master SPI mode operation.

- **Bit 2 – PERR: Parity Error**

If parity checking is enabled and the next character in the receive buffer has a parity error, this flag is set. If parity check is not enabled, this flag will always be read as zero. This bit is valid until the receive buffer (DATA) is read. Always write this bit location to zero when writing the STATUS register. For details on parity calculation, refer to “Parity Bit Calculation” on page 207.

This flag is not used in master SPI mode operation.

- **Bit 1 – Reserved**

This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.

- **Bit 0 – RXB8: Receive Bit 8**

RXB8 is the ninth data bit of the received character when operating with serial frames with nine data bits. When used, this bit must be read before reading the low bits from DATA.

This bit is unused in master SPI mode operation.

19.14.3 CTRLA – Control Register A

Bit	7	6	5	4	3	2	1	0
+0x03	–	–	RXCINTLVL[1:0]		TXCINTLVL[1:0]		DREINTLVL[1:0]	
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

- **Bit 7:6 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 5:4 – RXCINTLVL[1:0]: Receive Complete Interrupt Level**

These bits enable the receive complete interrupt and select the interrupt level, as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94. The enabled interrupt will be triggered when the RXCIF flag in the STATUS register is set.

- **Bit 3:2 – TXCINTLVL[1:0]: Transmit Complete Interrupt Level**

These bits enable the transmit complete interrupt and select the interrupt level, as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94. The enabled interrupt will be triggered when the TXCIF flag in the STATUS register is set.

- **Bit 1:0 – DREINTLVL[1:0]: Data Register Empty Interrupt Level**

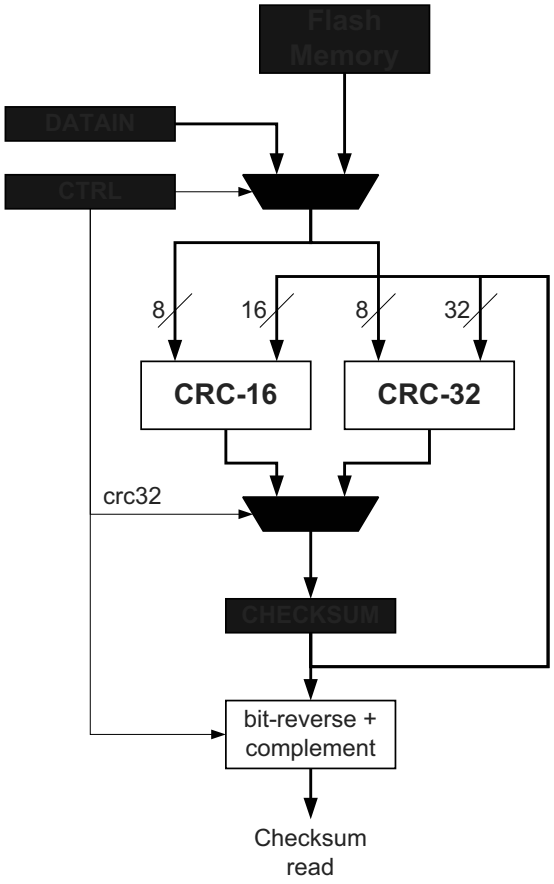
These bits enable the data register empty interrupt and select the interrupt level, as described in “Interrupts and Programmable Multilevel Interrupt Controller” on page 94. The enabled interrupt will be triggered when the DREIF flag in the STATUS register is set.

21.3 Operation

The data source for the CRC module must be selected in software as either flash memory or the I/O interface. The CRC module then takes data input from the selected source and generates a checksum based on these data. The checksum is available in the CHECKSUM registers in the CRC module. When CRC-32 polynomial is used, the final checksum read is bit reversed and complemented (see Figure 21-1).

For the I/O interface, which CRC polynomial is used is software selectable, but the default setting is CRC-16. CRC-32 is automatically used if Flash Memory is selected as the source. The CRC module operates on bytes only.

Figure 21-1. CRC Generator Block Diagram



21.4 CRC on Flash Memory

A CRC-32 calculation can be performed on the entire flash memory, on only the application section, on only the boot section, or on a software selectable range of the flash memory. Other than selecting the flash as the source, all further control and setup are done from the NVM controller. This means that the NVM controller configures the memory range to perform the CRC on, and the CRC is started using NVM commands. Once completed, the result is available in the checksum registers in the CRC module. For further details on setting up and performing CRC on flash memory, refer to “Memory Programming” on page 274.

21.5 CRC using the I/O Interface

CRC can be performed on any data by loading them into the CRC module using the CPU and writing the data to the DATAIN register. Using this method, an arbitrary number of bytes can be written to the register by the CPU, and CRC is done continuously for each byte. New data can be written for each cycle. The CRC complete is signaled by writing the BUSY bit in the STATUS register.

21.6 Register Description

21.6.1 CTRL – Control Register

Bit	7	6	5	4	3	2	1	0
+0x00	RESET[1:0]		CRC32	–	SOURCE[3:0]			
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7:6 – RESET[1:0]: Reset**

These bits are used to reset the CRC module, and they will always be read as zero. The CRC registers will be reset one peripheral clock cycle after the RESET[1] bit is set

Table 21-1. CRC Reset

RESET[1:0]	Group configuration	Description
00	NO	No reset
01	–	Reserved
10	RESET0	Reset CRC with CHECKSUM to all zeros
11	RESET1	Reset CRC with CHECKSUM to all ones

- **Bit 5 – CRC32: CRC-32 Enable**

Setting this bit will enable CRC-32 instead of the default CRC-16. It cannot be changed while the BUSY flag is set.

- **Bit 4 – Reserved**

This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.

- **Bit 3:0 – SOURCE[3:0]: Input Source**

These bits select the input source for generating the CRC. The selected source is locked until either the CRC generation is completed or the CRC module is reset. CRC generation complete is generated and signaled from the selected source when used with the flash memory.

Table 21-2. CRC Source Select

SOURCE[3:0]	Group configuration	Description
0000	DISABLE	CRC disabled
0001	IO	I/O interface
0010	FLASH	Flash
0011	–	Reserved for future use
0100	–	Reserved for future use
0101	–	Reserved for future use
0110	–	Reserved for future use
0111	–	Reserved for future use
1xxx	–	Reserved for future use

- **Bit 7:1 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 0 – IF: Interrupt Flag**

The interrupt flag is set when the ADC conversion is complete. If the channel is configured for compare mode, the flag will be set if the compare condition is met. IF is automatically cleared when the ADC channel interrupt vector is executed. The bit can also be cleared by writing a one to the bit location.

22.15.5 RESH – Result Register High

For all result registers and with any ADC result resolution, a signed number is represented in 2's complement form, and the msb represents the sign bit.

The RESL and RESH register pair represents the 16-bit value, ADCRESULT. Reading and writing 16-bit values require special attention. Refer to "Accessing 16-bit Registers" on page 11 for details.

	Bit	7	6	5	4	3	2	1	0
12-bit, left.		**RES[11:4]**							
12-bit, right	+0x05	–	–	–	–	**RES[11:8]**			
8-bit		–	–	–	–	–	–	–	–
	Read/Write	R	R	R	R	R	R	R	R
	Initial Value	0	0	0	0	0	0	0	0

22.15.5.1 12-bit Mode, Left Adjusted

- **Bit 7:0 – RES[11:4]: Channel Result High byte**

These are the eight msbs of the 12-bit ADC result.

22.15.5.2 12-bit Mode, Right Adjusted

- **Bit 7:4 – Reserved**

These bits will in practice be the extension of the sign bit, CHRES11, when the ADC works in differential mode, and set to zero when the ADC works in signed mode.

- **Bits 3:0 – RES[11:8]: Channel Result High bits**

These are the four msbs of the 12-bit ADC result.

22.15.5.3 8-bit Mode

- **Bit 7:0 – Reserved**

These bits will in practice be the extension of the sign bit, CHRES7, when the ADC works in signed mode, and set to zero when the ADC works in single-ended mode.

22.15.6 RESL – Result Register Low

	Bit	7	6	5	4	3	2	1	0
12-/8-bit, right	+0x04	**RES[7:0]**							
12-bit, left.		**RES[3:0]**				–	–	–	–
	Read/Write	R	R	R	R	R	R	R	R
	Initial Value	0	0	0	0	0	0	0	0

22.15.6.1 12-/8-bit Mode

- **Bit 7:0 – RES[7:0]: Result Low byte**

These are the eight lsbs of the ADC result.

- **Bit 1 – AC1IF: Analog Comparator 1 Interrupt Flag**

This is the interrupt flag for AC1. AC1IF is set according to the INTMODE setting in the corresponding “ACnCTRL – Analog Comparator n Control Register” on page 257.

This flag is automatically cleared when the analog comparator 1 interrupt vector is executed. The flag can also be cleared by writing a one to its bit location.

- **Bit 0 – AC0IF: Analog Comparator 0 Interrupt Flag**

This is the interrupt flag for AC0. AC0IF is set according to the INTMODE setting in the corresponding “ACnCTRL – Analog Comparator n Control Register” on page 257.

This flag is automatically cleared when the analog comparator 0 interrupt vector is executed. The flag can also be cleared by writing a one to its bit location.

23.8.7 CURRCTRL – Current Source Control Register

Bit	7	6	5	4	3	2	1	0
+0x08	**CURRENT**	–	–	–	–	–	**AC1CURR**	**AC0CURR**
Read/Write	R/W	R	R	R	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7 – CURRENT: Current Source Enable**

Setting this bit to one will enable the constant current source.

- **Bit 6:2 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 1 – AC1CURR: AC1 Current Source Output Enable**

Setting this bit to one will enable the constant current source output on the pin selected by MUXNEG in AC1MUXTRL.

- **Bit 0 – AC0CURR: AC0 Current Source Output Enable**

Setting this bit to one will enable the constant current source output on the pin selected by MUXNEG in AC0MUXTRL.

23.8.8 CURRCALIB – Current Source Calibration Register

Bit	7	6	5	4	3	2	1	0
+0x09	–	–	–	–	**CALIB[3:0]**			
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bits 7:4 – Reserved**

These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:0 – CALIB[3:0]: Current Source Calibration**

The constant current source is calibrated during production. A calibration value can be read from the signature row and written to the CURRCALIB register from software. Refer to device data sheet for default calibration values and user calibration range.

3. Execute the SPM instruction. This requires the timed CCP sequence during self-programming.

The BUSY flag in the STATUS register will be set until the operation is finished. The CPU will be halted during the complete execution of the command.

25.11.2.8 Erase Application Section / Boot Loader Section Page

The erase application section page erase and erase boot loader section page commands are used to erase one page in the application section or boot loader section.

1. Load the Z-pointer with the flash page address to erase. The page address must be written to ZPAGE. Other bits in the Z-pointer will be ignored during this operation.
2. Load the NVM CMD register with the erase application/boot section page command.
3. Execute the SPM instruction. This requires the timed CCP sequence during self-programming.

The BUSY flag in the NVM STATUS register will be set until the erase operation is finished. The FBUSY flag is set as long the flash is busy, and the application section cannot be accessed.

25.11.2.9 Application Section / Boot Loader Section Page Write

The write application section page and write boot loader section page commands are used to write the flash page buffer into one flash page in the application section or boot loader section.

1. Load the Z-pointer with the flash page to write. The page address must be written to FPAGE. Other bits in the Z-pointer will be ignored during this operation.
2. Load the NVM CMD register with the write application section/boot loader section page command.
3. Execute the SPM instruction. This requires the timed CCP sequence during self-programming.

The BUSY flag in the NVM STATUS register will be set until the write operation is finished. The FBUSY flag is set as long the flash is busy, and the application section cannot be accessed.

An invalid page address in the Z-pointer will abort the NVM command. The erase application section page command requires that the Z-pointer addresses the application section, and the erase boot section page command requires that the Z-pointer addresses the boot loader section.

25.11.2.10 Erase and Write Application Section / Boot Loader Section Page

The erase and write application section page and erase and write boot loader section page commands are used to erase one flash page and then write the flash page buffer into that flash page in the application section or boot loader section in one atomic operation.

1. Load the Z-pointer with the flash page to write. The page address must be written to FPAGE. Other bits in the Z-pointer will be ignored during this operation.
2. Load the NVM CMD register with the erase and write application section/boot loader section page command.
3. Execute the SPM instruction. This requires the timed CCP sequence during self-programming.

The BUSY flag in the NVM STATUS register will be set until the operation is finished. The FBUSY flag is set as long as the flash is busy, and the application section cannot be accessed.

An invalid page address in the Z-pointer will abort the NVM command. The erase and write application section command requires that the Z-pointer addresses the application section, and the erase and write boot section page command requires that the Z-pointer addresses the boot loader section.

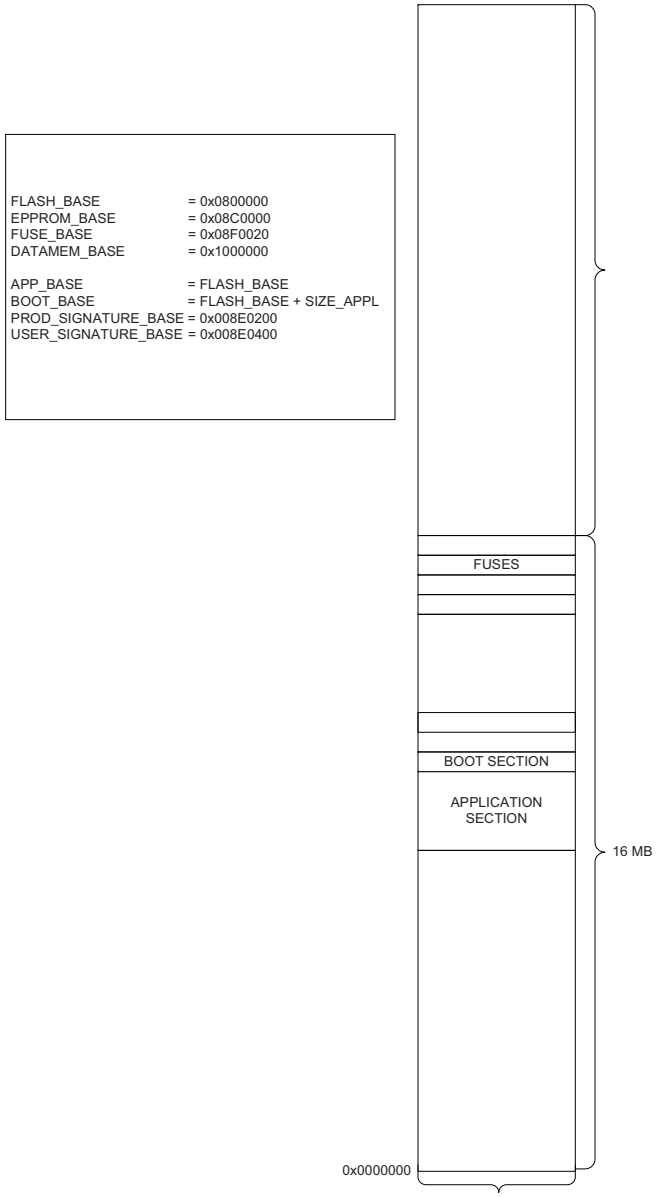
25.11.2.11 Application Section / Boot Loader Section CRC

The application section CRC and boot loader section CRC commands can be used to verify the application section and boot loader section content after self-programming.

1. Load the NVM CMD register with the application section/ boot load section CRC command.
2. Set the CMDEX bit in the NVM CTRLA register. This requires the timed CCP sequence during self-programming.

The BUSY flag in the NVM STATUS register will be set, and the CPU is halted during the execution of the CRC command. The CRC checksum will be available in the NVM data registers.

Figure 25-3. Memory Map for PDI Accessing the Data and Program Memories



25.12.1 Enabling External Programming Interface

NVM programming from the PDI requires enabling using the following steps:

1. Load the RESET register in the PDI with 0x59.
2. Load the NVM key in the PDI.
3. Poll NVMEN in the PDI status register (PDI STATUS) until NVMEN is set.

When the NVMEN bit in the PDI STATUS register is set, the NVM interface is enabled and active from the PDI.

25.12.2 NVM Programming

When the PDI NVM interface is enabled, all memories in the device are memory mapped in the PDI address space. The PDI controller does not need to access the NVM controller's address or data registers, but the NVM controller must be loaded with the correct command (i.e., to read from any NVM, the controller must be loaded with the NVM read command