Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

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

Applications of "Embedded - Microcontrollers"

Details	
Product Status	Obsolete
Core Processor	AVR
Core Size	8-Bit
Speed	4MHz
Connectivity	-
Peripherals	POR, WDT
Number of I/O	5
Program Memory Size	4KB (2K x 16)
Program Memory Type	FLASH
EEPROM Size	256 x 8
RAM Size	512 x 8
Voltage - Supply (Vcc/Vdd)	2V ~ 2.4V
Data Converters	A/D 1x10b
Oscillator Type	Internal
Operating Temperature	-20°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	18-WFDFN Exposed Pad (Staggered Leads)
Supplier Device Package	18-MLF (3.5x6.5)
Purchase URL	https://www.e-xfl.com/product-detail/microchip-technology/atmega4hvd-4mx

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

Assembly Code Example

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

C Code Example

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

6.7.1 Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is four clock cycles minimum. After four clock cycles the program vector address for the actual interrupt handling routine is executed. During this four clock cycle period, the Program Counter is pushed onto the Stack. The vector is normally a jump to the interrupt routine, and this jump takes three clock cycles. If an interrupt occurs during execution of a multi-cycle instruction, this instruction is completed before the interrupt is served. If an interrupt occurs when the MCU is in sleep mode, the interrupt execution response time is increased by four clock cycles. This increase comes in addition to the start-up time from the selected sleep mode.

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

7.3 SRAM Data Memory

Figure 7-2 shows how the ATmega4HVD/8HVD SRAM Memory is organized.

The ATmega4HVD/8HVD is a complex microcontroller with more peripheral units than can be supported within the 64 locations reserved in the 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.

The lower 512 data memory locations address both the Register File, the I/O memory, Extended I/O memory, and the internal data SRAM. The first 32 locations address the Register File, the next 64 location the standard I/O memory, then 160 locations of Extended I/O memory, and the next 512 locations address the internal data SRAM.

The five different addressing modes for the data memory cover: Direct, Indirect with Displacement, Indirect, Indirect with Pre-decrement, and Indirect with Post-increment. In the Register File, registers R26 to R31 feature the indirect addressing pointer registers.

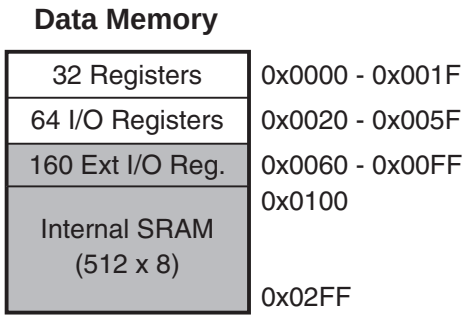
The direct addressing reaches the entire data space.

The Indirect with Displacement mode reaches 63 address locations from the base address given by the Y- or Z-register.

When using register indirect addressing modes with automatic pre-decrement and post-increment, the address registers X, Y, and Z are decremented or incremented.

The 32 general purpose working registers, 64 I/O Registers, 160 Extended I/O Registers, and the 512 bytes of internal data SRAM in the ATmega4HVD/8HVD are all accessible through all these addressing modes. The Register File is described in ”General Purpose Register File” on page 9.

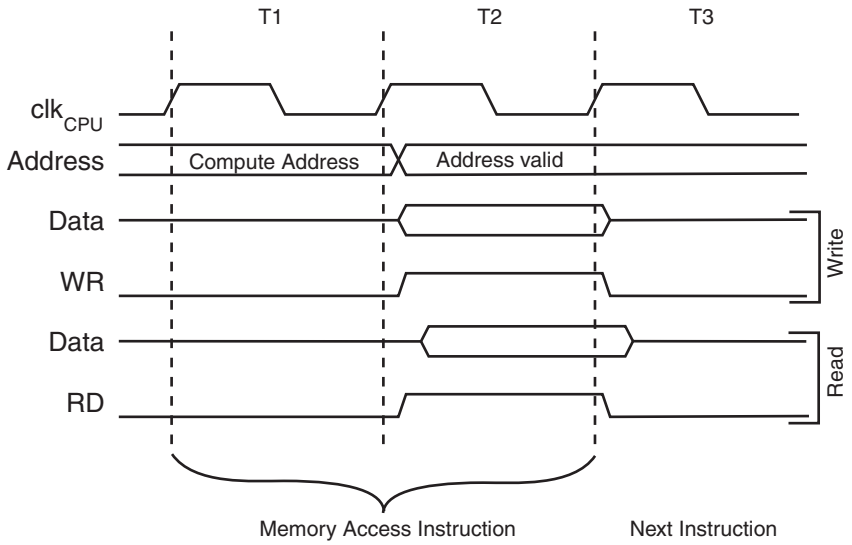
Figure 7-2. Data Memory Map



7.3.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. The internal data SRAM access is performed in two clk_{CPU} cycles as described in Figure 1.

Figure 1. On-chip Data SRAM Access Cycles



7.4 EEPROM Data Memory

The ATmega4HVD/8HVD contains 256 bytes of data EEPROM memory. It is organized as a separate data space, in which single bytes can be read and written. The EEPROM has an endurance of at least 100,000 write/erase cycles. The access between the EEPROM and the CPU is described in the following, specifying the EEPROM Address Registers, the EEPROM Data Register, and the EEPROM Control Register.

For a detailed description of Serial and Parallel data downloading to the EEPROM, see page 131 and page 131 respectively.

7.4.1 EEPROM Read/Write Access

The EEPROM Access Registers are accessible in the I/O space.

The write access time for the EEPROM is given in Table 7-1. A self-timing function, however, lets the user software detect when the next byte can be written. If the user code contains instructions that write the EEPROM, some precautions must be taken.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. Refer to the description of the EEPROM Control Register for details on this.

When the EEPROM is read, the CPU is halted for four clock cycles before the next instruction is executed. When the EEPROM is written, the CPU is halted for two clock cycles before the next instruction is executed.

7.5 I/O Memory

The I/O space definition of the ATmega4HVD/8HVD is shown in ”Register Summary” on page 151.

All ATmega4HVD/8HVD I/Os and peripherals are placed in the I/O space. All I/O locations may be accessed by the LD/LDS/LDD and ST/STS/STD instructions, transferring data between the 32 general purpose working registers and the I/O space. 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 instruc-

4. Write a logical one to the EEMPE bit while writing a zero to EEPE in EECR.
5. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

Caution:

An interrupt between step 4 and step 5 will make the write cycle fail, since the EEPROM Master Write Enable will timeout. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEARL or EEDR Register will be modified, causing the interrupted EEPROM access to fail. It is recommended to have the Global Interrupt Flag cleared during all the steps to avoid these problems.

When the write access time has elapsed, the EEPE bit is cleared by hardware. The user software can poll this bit and wait for a zero before writing the next byte. When EEPE has been set, the CPU is halted for two cycles before the next instruction is executed.

Note that a BLOD reset will abort any ongoing write operation and invalidate the result.

- **Bit 0 – EERE: EEPROM Read Enable**

The EEPROM Read Enable Signal EERE is the read strobe to the EEPROM. When the correct address is set up in the EEARL Register, the EERE bit must be written to a logic one to trigger the EEPROM read. The EEPROM read access takes one instruction, and the requested data is available immediately. When the EEPROM is read, the CPU is halted for four cycles before the next instruction is executed.

The user should poll the EEPE bit before starting the read operation. If a write operation is in progress, it is neither possible to read the EEPROM, nor to change the EEARL Register.

The Calibrated Fast RC Oscillator is used to time the EEPROM access and the programing time will therefore depend on the calibrated oscillator frequency. Table 7-2 lists the typical programming time for EEPROM access from the CPU.

Table 7-2. EEPROM Programming Time

Symbol	Number of Calibrated RC Oscillator Cycles	Typ Programming Time, f_{OSC} = 4.0 MHz
EEPROM write (from CPU)	13 600	3.4 ms

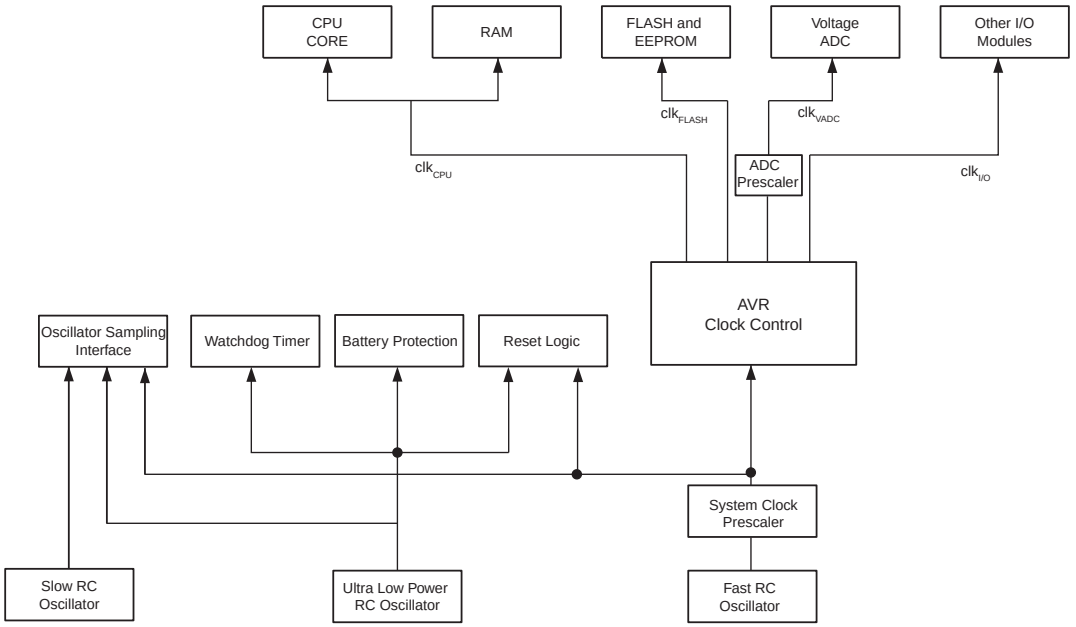
The following code examples show one assembly and one C function for writing to the EEPROM. The examples assume that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during execution of these functions. The examples also assume that no Flash Boot Loader is present in the software. If such code is present, the EEPROM write function must also wait for any ongoing SPM command to finish.

8. System Clock and Clock Options

8.1 Clock Systems and their Distribution

Figure 8-1 presents the principal clock systems in the AVR and their distribution. All of the clocks need not be active at a given time. In order to reduce power consumption, the clocks to modules not being used can be halted by using different sleep modes, as described in "Power Management and Sleep Modes" on page 32. The clock systems are detailed below.

Figure 8-1. Clock Distribution



8.1.1 CPU Clock – clk_{CPU}

The CPU clock is routed to parts of the system concerned with operation of the AVR core. Examples of such modules are the General Purpose Register File, the Status Register and the data memory holding the Stack Pointer. Halting the CPU clock inhibits the core from performing general operations and calculations.

8.1.2 I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules. The I/O clock is also used by the External Interrupt module, but note that some external interrupts are detected by asynchronous logic, allowing such interrupts to be detected even if the I/O clock is halted.

8.1.3 Flash Clock – clk_{FLASH}

The Flash clock controls operation of the Flash interface. The Flash clock is usually active simultaneously with the CPU clock.

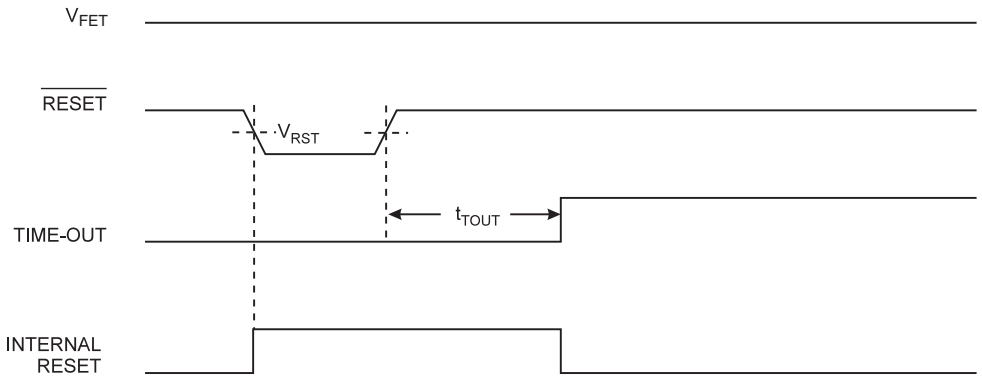
8.1.4 ADC Clock – clk_{ADC}

The ADC is provided with a dedicated clock domain. The ADC is automatically prescaled according to the System Clock Prescaler's setting to provide a fixed clock frequency to the

10.3 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 "System and Reset Characteristics" on page 144) 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 timeout period, t_{TOUT}, has expired.

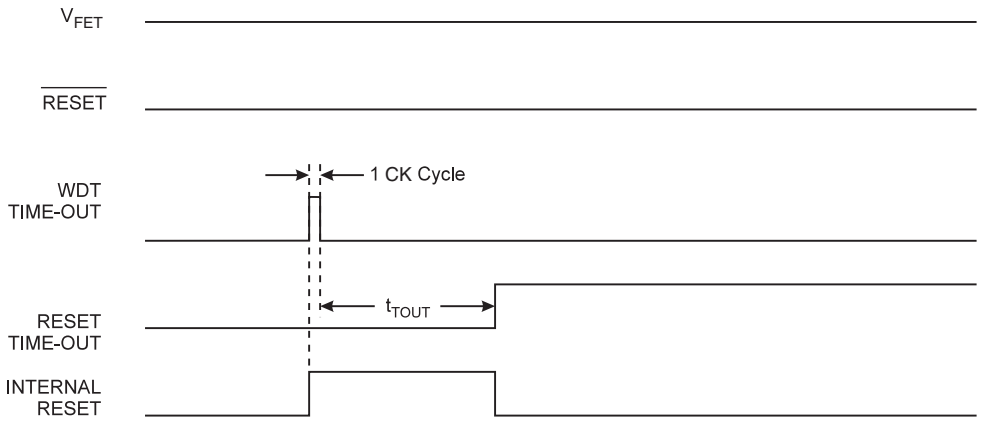
Figure 10-2. External Reset During Operation



10.4 Watchdog Reset

When the Watchdog times out, it will generate a short reset pulse of one CK cycle duration. On the falling edge of this pulse, the delay timer starts counting the timeout period t_{TOUT}. Refer to page 38 for details on operation of the Watchdog Timer.

Figure 10-3. Watchdog Reset During Operation



10.5 Black-out Detection

ATmega4HVD/8HVD has an on-chip Black-out Detection (BLOD) circuit for monitoring the VREG level during operation by comparing it to a trigger level defined by hardware.

The ATmega4HVD/8HVD has two detection levels and two application areas for BLOD, see "System and Reset Characteristics" on page 144. One detection level ($V_{BLOT, START-UP}$) is used to ensure that the voltage on VFET is sufficient to operate the voltage regulator within its specifications when the chip starts up. The other detection level is used during normal operation ($V_{BLOT, NORMAL}$) to determine if VREG drops below a voltage where correct operation cannot be

Note: If the Watchdog is accidentally enabled, for example by a runaway pointer or Black-out condition, the device will be reset and the Watchdog Timer will stay enabled. If the code is not set up to handle the Watchdog, this might lead to an eternal loop of timeout resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialisation routine, even if the Watchdog is not in use.

The following code example shows one assembly and one C function for changing the timeout value of the Watchdog Timer.

Assembly Code Example[1]

```
WDT_Prescaler_Change:
  ; Turn off global interrupt
  cli
  ; Reset Watchdog Timer
  wdr
  ; Start timed sequence
  in    r16, WDTCSR
  ori   r16, (1<<WDCE) | (1<<WDE)
  out   WDTCSR, r16
  ; --  Got four cycles to set the new values from here -
  ; Set new prescaler(timeout) value = 64K cycles (~0.5 s)
  ldi   r16, (1<<WDE) | (1<<WDP2) | (1<<WDP0)
  out   WDTCSR, r16
  ; --  Finished setting new values, used 2 cycles -
  ; Turn on global interrupt
  sei
  ret
```

C Code Example[1]

```
void WDT_Prescaler_Change(void)
{
  __disable_interrupt();
  __watchdog_reset();
  /* Start timed  equence */
  WDTCSR |= (1<<WDCE) | (1<<WDE);
  /* Set new prescaler(timeout) value = 64K cycles (~0.5 s) */
  WDTCSR  = (1<<WDE) | (1<<WDP2) | (1<<WDP0);
  __enable_interrupt();
}
```

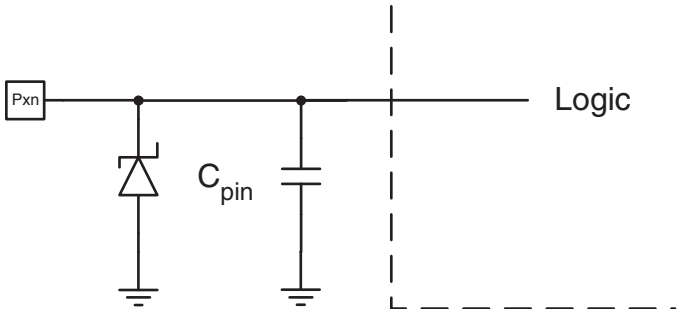
Note: 1. See "About Code Examples" on page 5.

Note: The Watchdog Timer should be reset before any change of the WDP bits, since a change in the WDP bits can result in a timeout when switching to a shorter timeout period.

13. High Voltage I/O Ports

All high voltage AVR ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the state of one port pin can be changed without unintentionally changing the state of any other pin with the SBI and CBI instructions. All high voltage I/O pins have protection Zener diodes to Ground as indicated in Figure 13-1. See ”Electrical Characteristics” on page 142 for a complete list of parameters.

Figure 13-1. High Voltage I/O Pin Equivalent Schematic



Note: 1. See Figure 13-2 on page 57 for details.

All registers and bit references in this section are written in general form. A lower case “x” represents the numbering letter for the port, and a lower case “n” represents the bit number. However, when using the register or bit defines in a program, the precise form must be used. For example, PORTC3 for bit number three in Port C, here documented generally as PORTxn. The physical I/O Registers and bit locations are listed in ”Register Description” on page 60.

One I/O Memory address location is allocated for each high voltage port, the Data Register – PORTx. The Data Register is read/write.

Using the I/O port as General Digital Output is described in ”High Voltage Ports as General Digital Outputs” on page 57.

13.2.1 Alternate Functions of Port C

The Port C pins with alternate functions are shown in Table 13-2.

Table 13-2. Port C Pins Alternate Functions

Port Pin	Alternate Function
PC0	INT0/ICP0/XTAL (External Interrupt 0, Timer/Counter 0 input Capture Trigger or External Clock)
PC1	MOSI/INT1/EXT_PROT (SPI BUS Serial Data Input, External Interrupt 1, External Protection Input)

The alternate pin configuration is as follows:

- **INT0/ICP0/XTAL - Port C, Bit 0**

INT0, External Interrupt 0: When INT0 is written to one and the I-bit in the Status Register (SREG) is set (one), the corresponding external pin interrupt is enabled. The Interrupt Sense Control bits in the "EICRA – External Interrupt Control Register A" on page 53 - defines whether the external interrupt is activated on rising or falling edge or level sensed. Activity on any of these pins will trigger an interrupt request even if the pin is enabled as an output. This provides a way of generating a software interrupt.

XTAL, External Clock: When the CKSEL fuse is programmed, PC0 is used as clock source instead of the Internal RC oscillator (For test purposes only).

- **MOSI/INT1/EXT_PROT - Port C, Bit 1**

MOSI, Slave Data Input pin for SPI Programming.

INT1, External Interrupt 1: When INT1 is written to one and the I-bit in the Status Register (SREG) is set (one), the corresponding external pin interrupt is enabled. The Interrupt Sense Control bits in the "EICRA – External Interrupt Control Register A" on page 53 - defines whether the external interrupt is activated on rising or falling edge or level sensed. Activity on any of these pins will trigger an interrupt request even if the pin is enabled as an output. This provides a way of generating a software interrupt.

EXT_PROT, External Protection Input: When the EPID bit in the BPCR Register is cleared, the External Protection Input functionality is enabled. Note that this port overriding is default enabled.

Table 13-3 relates the alternate functions of Port C to the overriding signals shown in Figure 13-3 on page 58.

Table 13-3. Overriding Signals for Alternate Functions in PC1:0

Signal Name	PC1/MOSI/INT1/EXT_PROT	PC0/INT0/ICP0/XTAL
PVOE	0	0
PVOV	0	0
DIEOE	INT Enable + $\overline{\text{EPID}}$	INT Enable + $\overline{\text{CKSEL}}$
DIEOV	1	1
DI	INT1/EXT_PROT	INT0/ICP0/XTAL INPUT

16.2.1 Registers

The Timer/Counter Low Byte Register (TCNTnL) and Output Compare Registers (OCRnA and OCRnB) are 8-bit registers. Interrupt request (abbreviated to Int.Req. in Figure 16-1 on page 74) signals are all visible in the Timer Interrupt Flag Register (TIFR). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK). TIFR and TIMSK are not shown in the figure.

In 16-bit mode the Timer/Counter consists one more 8-bit register, the Timer/Counter High Byte Register (TCNTnH). Furthermore, there is only one Output Compare Unit in 16-bit mode as the two Output Compare Registers, OCRnA and OCRnB, are combined to one 16-bit Output Compare Register. OCRnA contains the low byte of the word and OCRnB contains the higher byte of the word. When accessing 16-bit registers, special procedures described in section "Accessing Registers in 16-bit Mode" on page 82 must be followed.

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 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}).

16.2.2 Definitions

Many register and bit references in this section are written in general form. A lower case "n" replaces the module number, e.g. Timer/Counter number. A lower case "x" replaces the unit, e.g. OCRnx and ICPnx describes OCRnA/B and ICP1/0x . However, when using the register or bit defines in a program, the precise form must be used, i.e., TCNT0L for accessing Timer/Counter0 counter value and so on.

The definitions in Table 16-1 are also used extensively throughout the document.

Table 16-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255) in 8-bit mode or 0xFFFF (decimal 65535) in 16-bit mode.
TOP	The counter reaches the TOP when it becomes equal to the highest value in the count sequence. The TOP value can be assigned to be the fixed value 0xFF/0xFFFF (MAX) or the value stored in the OCRnA Register.

16.3 Timer/Counter Clock Sources

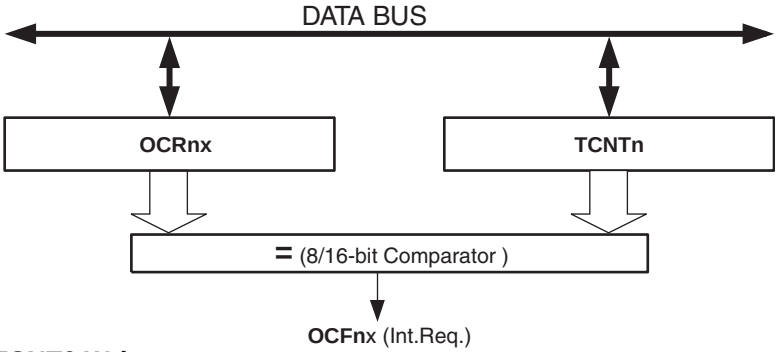
The Timer/Counter can be clocked internally, via the prescaler, or by an external clock source. The Clock Select logic is controlled by the Clock Select (CSn2:0) bits located in the Timer/Counter Control Register n B (TCCRnB), and controls which clock source and edge the Timer/Counter uses to increment 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}). For details on clock sources and prescaler, see "Timer/Counter0 and Timer/Counter1 Prescalers" on page 71

16.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 16-2 on page 76 shows a block diagram of the counter and its surroundings.

OCFnA as there is only one Output Compare Unit. If the corresponding interrupt is enabled, the Output Compare Flag generates an Output Compare interrupt. The Output Compare Flag is automatically cleared when the interrupt is executed. Alternatively, the flag can be cleared by software by writing a logical one to its I/O bit location. Figure 16-5 on page 81 shows a block diagram of the Output Compare unit.

Figure 16-5. Output Compare Unit, Block Diagram



16.7.1 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNTnH/L Register will block any Compare Match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCRnA/B to be initialized to the same value as TCNTn without triggering an interrupt when the Timer/Counter clock is enabled.

16.7.2 Using the Output Compare Unit

Since writing TCNTnH/L will block all Compare Matches for one timer clock cycle, there are risks involved when changing TCNTnH/L when using the Output Compare Unit, independently of whether the Timer/Counter is running or not. If the value written to TCNTnH/L equals the OCRnA/B value, the Compare Match will be missed.

16.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{Tn}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set. Figure 16-6 on page 81 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value.

Figure 16-6. Timer/Counter Timing Diagram, no Prescaling

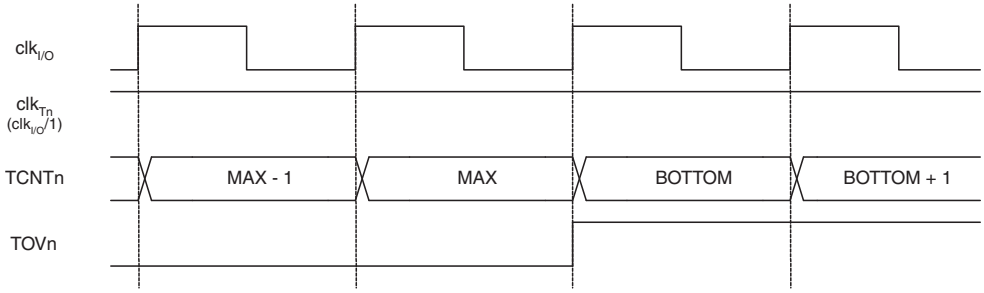


Figure 16-7 on page 82 shows the same timing data, but with the prescaler enabled.

17. ADC - Analog-to-Digital Converter

17.1 Features

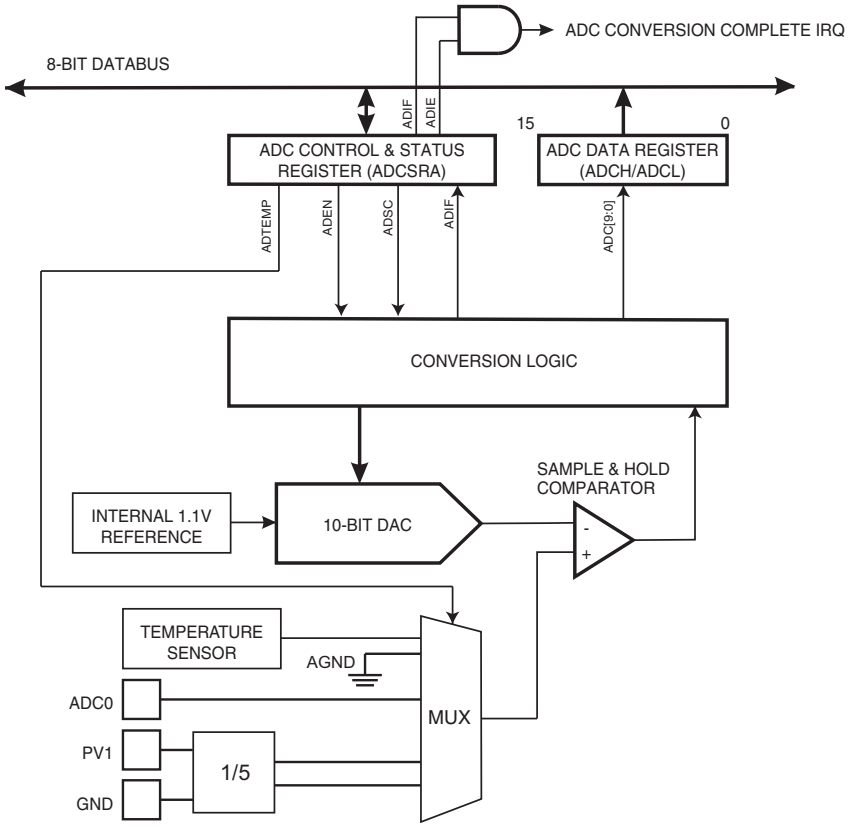
- **10-bit Resolution**
- **78 µs Conversion Time @ clk_{ADC} = 167 kHz**
- **Up to 13 kSPS at Maximum Resolution**
- **External Input Channel with 0 - 5V Input Voltage Range**
- **External Input Channel (ADC0) with 0 - 1V Input Voltage Range**
- **Internal Temperature Sensor Input Channel**
- **1.1V ADC Reference Voltage (typical value)**
- **Interrupt on ADC Conversion Complete**
- **Sleep Mode Noise Canceler**

The ATmega4HVD/8HVD features a 10-bit successive approximation ADC. The single-ended voltage inputs refer to 0V.

The ADC contains a Sample and Hold circuit which ensures that the input voltage to the ADC is held at a constant level during conversion. A block diagram of the ADC is shown in Figure 17-1.

Internal reference voltage of nominally 1.1V is provided on-chip. External input on PV1 pin is divided by 5 internally to be within the range of the internal reference votlage.

Figure 17-1. Analog-to-Digital Converter Block Schematic



19.3 Battery Pack Short mode

The Voltage Regulator has a separate Short-Circuit Detection mode (RSCD) that can be enabled or disabled by SW. This mode should always be enabled except when operating at VFET voltages below V_{FORCE} (see TBD-electrical chara). The mode is intended for sustaining operation during short spikes on VFET that can occur for instance during a battery pack insertion. The mode is entered when VFET drops below V_{FORCE} and Regulator Short-circuit Detection is enabled. In the Battery Pack Short mode, VFET is temporarily disconnected from VREG to avoid a quick drop in the voltage regulator output. The chip will be completely powered by the external reservoir capacitor (CREG). This allows the chip to operate a certain time before entering BLOD reset (power-off) even if the VFET voltage is too low for the voltage regulator to operate.

The maximum time that the chip can operate in the Battery Pack Short mode is given by the size of the external reservoir capacitor and the actual power drawn from VREG. The VREG voltage must stay above $V_{BLOT,normal}$ to avoid that the chip enters power-off. If a battery pack short occurs when VREG = 2.2V and $V_{BLOT,\ normal}$ is 2.0V the chip can operate for a time given by:

$$T = \frac{c \cdot \Delta(v)}{I_{AVG}} = \frac{CREG \cdot 0.2V}{I_{AVG}}$$

where I_{AVG} represents the average current drawn from CREG. For CREG = 2.2 µF and I_{AVG} = 100 µA, this time equals 4.4 ms. Refer to Table 19-2 on page 102 for an example of Regulator Short-circuit Detection.

19.4 Regulator Force mode

The regulator Force mode is designed to be able to operate the chip even at very low cell voltages. This mode is automatically entered when the VFET voltage drops below V_{FORCE-} provided that the Regulator Short-circuit Detection is disabled. To ensure operation down to minimum VFET level, Regulator Short-circuit Detection should always be disabled before VFET reaches V_{FORCE-} during discharge.

An example of VREG voltage as function of VFET when using the voltage regulator as intended, is shown in figure 66. When VFET > V_{FORCE}, VREG is regulated to 2.2V (nominal value). When VFET approaches V_{FORCE}, Regulator Short-circuit Detection is disabled, and when VFET passes V_{FORCE-}, the regulator enters FORCE mode. When this occurs, a discontinuity in VREG can be observed. This is caused by the fact that VREG is no longer regulated and is forced as close to VFET as possible. A small difference V_{DROP} between VFET and VREG can be observed. This voltage drop depends on the load current. Typical vales can be found in TBD-El.Chara. The chip will continue operation in FORCE mode until VREG reaches the BLOD level and the chip enters power-off.

When using VREG as reference for external measurements, for instance as reference for an external thermistor as shown in TBD-operating circuit, it may be required to know the accurate value of VREG. In FORCE mode, VREG will range from V_{FORCE} to $VFET_{min}$, as opposed to normal regulation where VREG has a constant value. Since ATmega4/8HVD has no method of measuring VREG directly, it is recommended to measure the cell voltage and estimate VREG based on loss through Charge FET (if applied) and voltage drop from VFET to VREG according to the following formula:

$$VREG_{FORCE} = V_{Cell} - V_{DS-CFET} - V_{DROP}$$

20. Battery Protection

20.1 Features

- **Short-circuit Protection**
- **Discharge Over-current Protection**
- **Charge Over-current Protection**
- **External Protection Input**
- **Programmable and Lockable Detection Levels and Reaction Times**
- **Autonomous Operation Independent of CPU**

20.2 Overview

The Current Battery Protection circuitry (CBP) monitors the charge and discharge current and disables C-FET and D-FET if a Short-circuit, Over-current or High-current condition is detected. There are three different programmable detection levels: Short-circuit Detection Level, Discharge Over-current Detection Level, and Charge Over-current Detection Level. There are two different programmable delays for activating Current Battery Protection: Short-circuit Reaction Time and Over-current Reaction Time. After Current Battery Protection has been activated, the application software must re-enable the FETs. The Battery Protection hardware provides a hold-off time of 1 second nominally before software can re-enable the discharge FET. This provides safety in case the application software should unintentionally re-enable the discharge FET too early.

The activation of a protection also issues an interrupt to the CPU. The battery protection interrupts can be individually enabled and disabled by the CPU.

In addition, the module offers an External Protection Input. The activation of the External Protection Input operates independently of the rest of the battery protection mechanisms. The activation/deactivation of this protection is instantaneously controlled from the External Protection Input port, and will not deactivate or affect the other battery protection mechanisms.

The effect of the various battery protection types is given in Table 20-1.

Table 20-1. Effect of Battery Protection Types

Battery Protection Type	Interrupt Requests	C-FET	D-FET	MCU
Short-circuit Protection	Entry	Disabled	Disabled	Operational
Discharge Over-current Protection	Entry	Disabled	Disabled	Operational
Charge Over-current Protection	Entry	Disabled	Disabled	Operational
External Protection Input	Entry and/or Exit	Disabled	Disabled	Operational

In order to reduce power consumption, Short-circuit and Discharge Over-current Protection are automatically deactivated when the D-FET is disabled. The Charge Over-current is disabled when the C-FET is disabled. Note that Charge Over-current Protection is never automatically disabled when the chip is operated in DUVR mode. Also note that none of the current protections are deactivated by External Protection Input. To save power during an External Protection event, DFE and CFE should be cleared in the FCSR register. Make also sure that the chip is not operated in DUVR mode.

20.6 External Protection Input

The External Protection Input disables both FETs (Charge FET and Discharge FET) immediately (asynchronously) when the voltage on PC1 is pulled high (logic ‘1’) by the External Protection circuitry. It is also used to disable DUVR mode if DUVR mode is enabled. Note that, unlike a Battery Protection event, the External Protection input does not affect the status of the FCSR (CFE, DFE, DUVRD) bits. When the ‘high’ condition disappears, the FET disabling is released immediately. DUVR mode is automatically re-entered if enabled.

The feature is automatically enabled when the chip starts up, and can be disabled before locking the BPCR register. When locking the BPCR register, the External Protection feature is also locked. The feature should be disabled if it is not used.

When External Protection Input is enabled, an override enable signal is set to PC1, configuring the pin as digital input. The port may be set up to give an interrupt when the pin value changes. The protection status can be read from the port register.

Note that the External Protection Input is default enabled. This means that after reset (and during reset) the port is default overridden to External Protection Input, independent of the port register setting. The user must disable the External Protection Input before the port can be used as a normal port.

It is recommended that the external interrupt on the External Protection Input port (PC1) is configured to ‘any edge’ to generate an interrupt to the microcontroller when using this feature, indicating that the FET protection status has changed. Refer to ”External Interrupt” on page 53. By reading the pin register, the External Protection status can be determined. If the pin register is set, it means that External Protection is triggered and the FET control signals (FCSR (CFE, DFE and DUVRD)) are overridden so the FETs are disabled. In the opposite case External Protection violation is not present.

To ensure a safe exit from the External Protection Input condition, the FETs and DUVR mode should be disabled by SW when an External Protection condition is detected. This enables software to control completely when the FETs are switched ON again.

If not disabled by SW, the FETs will be re-enabled once the External Protection condition disappears (PC1 pin equals '0'). This feature may be useful to filter out glitches on the PC1 pin, for instance caused by temporary high voltages on the BATT pin when connecting a charger (refer to ”Operating Circuit” on page 141). In this case, SW does not have to take any action to re-enable the FETs.

Note however that compared to SW switching on the FETs, the switch ON time for the FETs is multiplied by a factor 10 when the FETs are disabled and re-enabled by External Protection Input and the chip is operated in sleep mode. This is caused by the internal clock system and power saving functions of the chip. For this reason, if the switch ON time of the transistors is critical, External Interrupt on PC1 should always be enabled when using the External Protection Input feature. The interrupt should be configured to trig on any edge and will ensure that the chip wakes up from sleep and thus enables a normal switch ON time for the FETs. To ensure a fast rise time on the OC and OD pins, SW must either remain in ACTIVE or IDLE mode during FET charging or disable and re-enable the OC/OD bits before entering Power-save sleep mode. For an example on correct External Protection Input usage, refer to Figure 20-1.

20.8.6 BPDOCD – Battery Protection Discharge-Over-current Detection Level Register

Bit	7	6	5	4	3	2	1	0	
	DOCDL[7:0]								BPDOCD
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	1	

- **Bits 7:0 – DOCDL7:0: Discharge Over-current Detection Level**

These bits sets the R_{SENSE} voltage level for detection of Discharge Over-current, as defined in Table 20-4 on page 112. This register should always be written as one-hot.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPDOCD register is written. Any writing to the BPDOCD register during this period will be ignored.

20.8.7 BPCOCD – Battery Protection Charge-Over-current Detection Level Register

Bit	7	6	5	4	3	2	1	0	
	COCDL[7:0]								BPCOCD
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	1	

- **Bits 7:0 –COCDL7:0: Charge Over-current Detection Level**

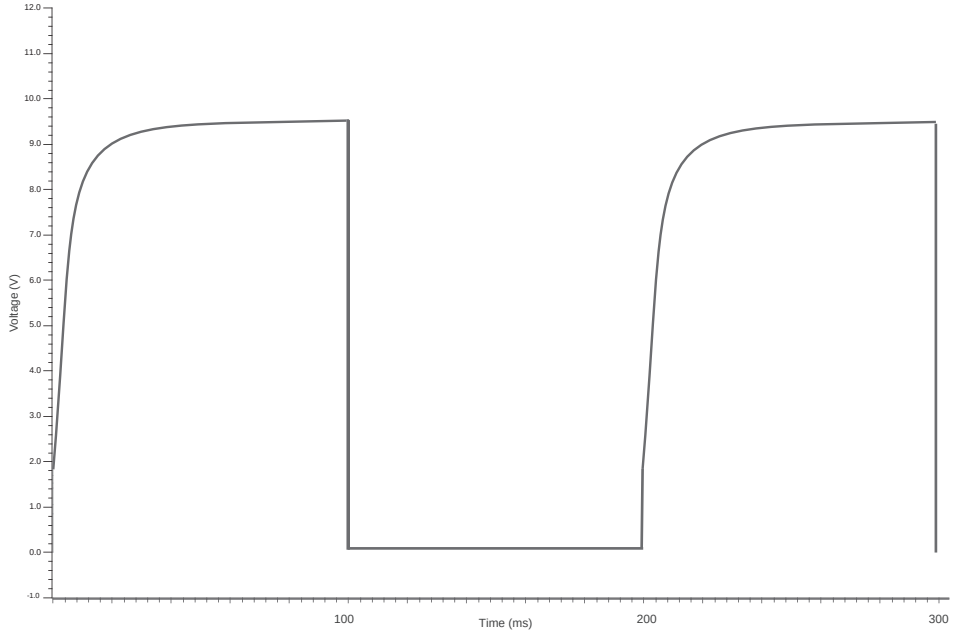
These bits sets the R_{SENSE} voltage level for detection of Charge Over-current, as defined in Table 20-4 on page 112. This register should always be written as one-hot.

Note: Due to synchronization of parameters between clock domains, a guard time of 3 ULP oscillator cycles + 3 CPU clock cycles is required between each time the BPCOCD register is written. Any writing to the BPCOCD register during this period will be ignored.

Table 20-4. DL[7:0] with corresponding R_{SENSE} Current for all Current Detection Levels (R_{SENSE} = 10 mΩ, VREF = 1.100 ± 0.005V)

Current Protection Detection Level			
DL[7:0]	**Min.**	**Typ.**	**Max.**
0x01	0.5A	2.0A	3.5A
0x02	1.0A	2.5A	4.0A
0x04	1.5A	3.0A	4.5A
0x08	2.0A	3.5A	5.0A
0x10	2.5A	4.0A	5.5A
0x20	3.0A	4.5A	6.0A
0x40	3.5A	5.0A	6.5A
0x80	4.5A	6.0A	7.5A
All other values	Reserved		

Figure 21-3. Switching NFET on and off during NORMAL operation



24.8 High-voltage Serial Programming Algorithm

To program and verify the ATmega4HVD/8HVD in the High-voltage Serial Programming mode, the following sequence is recommended (See instruction formats in Table 24-14):

24.8.1 Enter High-voltage Serial Programming Mode

The following algorithm puts the device in (High-voltage) Serial Programming mode:

1. Set Prog_enable pins listed in Table 24-12 on page 135 to “0000”, RESET pin to 0V and V_{CC} to 0V. VFET should not be connected.
2. Apply 3.0 - 3.5V between V_{CC} and GND, and between BATT and GND. Ensure that V_{CC} reaches at least 1.8V within the next 20 µs.
3. Wait 20 - 60 µs, and apply V_{HRST} - 12.5V to RESET.
4. Keep the Prog_enable pins unchanged for at least t_{HVRST} after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Release Prog_enable[1] pin to avoid drive contention on the Prog_enable[1]/SDO pin.
6. Wait at least 300 µs before giving any serial instructions on SDI/SII.

If the rise time of the V_{CC} is unable to fulfill the requirements listed above, the following alternative algorithm can be used.

1. Set Prog_enable pins listed in Table 24-12 on page 135 to “0000”, RESET pin to 0V, V_{CC} to 0V. VFET should not be connected.
2. Apply 3.0 - 3.5V between V_{CC} and GND, and between BATT and GND.
3. Monitor V_{CC}, and as soon as V_{CC} reaches 0.9 - 1.1V, apply V_{HRST} - 12.5V to RESET.
4. Keep the Prog_enable pins unchanged for at least t_{HVRST} after the High-voltage has been applied to ensure the Prog_enable Signature has been latched.
5. Release Prog_enable[1] pin to avoid drie contention on the Prog_enable[1]/SDO pin.
6. Wait until V_{CC} actually reaches 3.0 - 3.5V before giving any serial instructions on SDI/SII.

Table 24-13. High-voltage Reset Characteristics

Supply Voltage	RESET Pin High-voltage Threshold	Minimum High-voltage Period for Latching Prog_enable
V_{CC}	V_{HVRST}	t_{HVRST}
3.0V	11.5V	10 µs
3.5V	11.5V	10 µs

24.8.2 Considerations for Efficient Programming

The loaded command and address are retained in the device during programming. For efficient programming, the following should be considered.

- The command needs only be loaded once when writing or reading multiple memory locations.
- Skip writing the data value 0xFF that is the contents of the entire EEPROM (unless the EESAVE Fuse is programmed) and Flash after a Chip Erase.
- Address High byte needs only be loaded before programming or reading a new 256 word window in Flash or 256 byte EEPROM. This consideration also applies to Signature bytes reading.

Table 24-14. High-voltage Serial Programming Instruction Set for ATmega4HVD/8HVD (Continued)

Instruction		Instruction Format				Operation Remarks
		Instr.1/5	Instr.2/6	Instr.3	Instr.4	
Read EEPROM Byte	SDI	0_**bbbb_bbbb**_00	0_**aaaa_aaaa**_00	0_0000_0000_00	0_0000_0000_00	
	SII	0_0000_1100_00	0_0001_1100_00	0_0110_1000_00	0_0110_1100_00	
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	**q_qqqq_qqq**0_00	
Write Fuse Low Bits	SDI	0_0100_0100_00	0_**A987_6543**_00	0_0000_0000_00	0_0000_0000_00	Wait after Instr. 4 until SDO goes high. Write **A - 3** = “0” to program the Fuse bit.
	SII	0_0100_1100_00	0_0010_1100_00	0_0110_0100_00	0_0110_1100_00	
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	
Write Lock Bits	SDI	0_0010_0000_00	0_0000_00**21**_00	0_0000_0000_00	0_0000_0000_00	Wait after Instr. 4 until SDO goes high. Write **2 - 1** = “0” to program the Lock Bit.
	SII	0_0100_1100_00	0_0010_1100_00	0_0110_0100_00	0_0110_1100_00	
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	
Read Fuse Low Bits	SDI	0_0000_0100_00	0_0000_0000_00	0_0000_0000_00		Reading **A - 3** = “0” means the Fuse bit is programmed.
	SII	0_0100_1100_00	0_0110_1000_00	0_0110_1100_00		
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	**A_9876_543**x_xx		
Read Lock Bits	SDI	0_0000_0100_00	0_0000_0000_00	0_0000_0000_00		Reading **2, 1** = “0” means the Lock bit is programmed.
	SII	0_0100_1100_00	0_0111_1000_00	0_0111_1100_00		
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_x**21**x_xx		
Read Signature Bytes	SDI	0_0000_1000_00	0_0000_00**bb**_00	0_0000_0000_00	0_0000_0000_00	Repeats Instr 2 4 for each signature byte address.
	SII	0_0100_1100_00	0_0000_1100_00	0_0110_1000_00	0_0110_1100_00	
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	**q_qqqq_qqq**x_xx	
Read Calibration Byte	SDI	0_0000_1000_00	0_0000_0000_00	0_0000_0000_00	0_0000_0000_00	
	SII	0_0100_1100_00	0_0000_1100_00	0_0111_1000_00	0_0111_1100_00	
	SDO	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	x_xxxx_xxxx_xx	**p_pppp_ppp**x_xx	
Load “No Operation” Command	SDI	0_0000_0000_00				
	SII	0_0100_1100_00				
	SDO	x_xxxx_xxxx_xx				

Note: **a** = address high bits, **b** = address low bits, **d** = data in high bits, **e** = data in low bits, **p** = data out high bits, **q** = data out low bits, x = don’t care, **1** = Lock Bit1, **2** = Lock Bit2, **3** = CKSEL Fuse, **4** = SUT0 Fuse, **5** = SUT1 Fuse, Fuse, **A** = WDTON Fuse, **9** = EESAVE Fuse, **8** = SPIEN Fuse, **7** = DWEN Fuse, **6** = SELFPRGEN Fuse

Notes:
1. For page sizes less than 256 words, parts of the address (bbbb_bbbb) will be parts of the page address.
2. The EEPROM is written page-wise. But only the bytes that are loaded into the page are actually written to the EEPROM. Page-wise EEPROM access is more efficient when multiple bytes are to be written to the same page. Note that auto-erase of EEPROM is not available in High-voltage Serial Programming, only in SPI Programming.

Table 26-1. Electrical Characteristics[1], T_A = -20°C to 85°C, VFET = 2.4 to 4.2V (unless otherwise noted) (Continued)

	Parameter	Condition	Min	Typ	Max	Unit
Voltage Regulator	Regulated Output Voltage, VREG (Linear regulation mode)	VFET = 3.0	2.0	2.2	2.4	V
	Regulator Output Voltage, VREG (Force mode)			VFET - V_{DROP}	V_{FORCE}[2]	V
	Operating Voltage, VFET	Active mode, EEPROM writing	2.3		6.0	V
		Active mode, no EEPROM writing	2.1		6.0	V
	Regulator Force mode level (V_{FORCE})	Rising edge		2.50	2.7	V
		Falling edge		2.35	2.55	V
	Regulator Force level hysteresis			150		mV
	Regulator drop in Force mode (VFET - VREG)	VFET = 2.2V, I_{load} = 1 mA		50		

Notes: 1. All Electrical Characteristics contained in this data sheet are based on initial characterization of actual silicon. These values are preliminary values.
2. TBD

26.2 Oscillator Characteristics

Table 2. Oscillator Characteristics, T_A = -20°C to 85°C, V_{CC} = 2.2V (unless otherwise noted)

	Parameter	Condition	Min	Typ	Max	Unit
Slow RC Oscillator	Frequency		91	131	171	kHz
	Frequency Prediction Error[1]				1	%
	Frequency drift as function of V_{CC} [1][2]	$2.1 \le V_{CC} \le 2.7$		4		
ULP RC Oscillator	Frequency		89	128	167	kHz
Fast RC Oscillator	After initial calibration		1.9	2.0	2.1	MHz
	FOSCCAL frequency step size		0.5	1	2	%
	Frequency drift as function of V_{CC} [1][2]	$2.1 \le V_{CC} \le 2.7$, T_A = 25°C		0.5		%

Notes: 1. Not tested in production.
2. Relevant if regulatoris operated in Force mode.