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	8KB (4K 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/atmega8hvd-4mx

tains a high-voltage tolerant, open-drain IO pin that supports serial communication. Programming can be done in-system using the 4 General Purpose IO ports that support SPI programming

The MCU includes 4K/8K bytes of In-System Programmable Flash with Self-programming capabilities, 256 bytes EEPROM, 512 bytes SRAM, 32 general purpose working registers, 4 general purpose I/O lines, debugWIRE for On-chip debugging and SPI for In-system Programming, two flexible Timer/Counters with Input Capture, internal and external interrupts, a 10-bit ADC for measuring the cell voltage and on-chip temperature, a programmable Watchdog Timer with wake-up capabilities, and software selectable power saving modes.

The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The device is manufactured using Atmel's high voltage high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System, by a conventional non-volatile memory programmer or by an On-chip Boot program running on the AVR core.

The ATmega4HVD/8HVD AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, and On-chip Debugger.

The ATmega4HVD/8HVD is a low-power CMOS 8-bit microcontroller based on the AVR architecture. It is part of the AVR Smart Battery family that provides secure authentication, highly accurate monitoring and autonomous protection for Lithium-ion battery cells.

3. Resources

A comprehensive set of development tools, application notes and datasheets are available for download on http://www.atmel.com/avr.

4. Data Retention

Reliability Qualification results show that the projected data retention failure rate is much less than 1 PPM over 20 years at 85°C or 100 years at 25°C.

5. About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. These code examples assume that the part specific header file is included before compilation. Be aware that not all C compiler vendors include bit definitions in the header files and interrupt handling in C is compiler dependent. Please confirm with the C compiler documentation for more details.

For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instru0ctions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

tions. Refer to the instruction set section for more details. When using the I/O specific commands IN and OUT, the I/O addresses 0x00 - 0x3F must be used. When addressing I/O Registers as data space using LD and ST instructions, 0x20 must be added to these addresses. The ATmega4HVD/8HVD is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in Opcode for the IN and OUT instructions. For the Extended I/O space from 0x60 - 0xFF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits should be written to zero if accessed. Reserved I/O memory addresses should never be written.

Some of the status flags are cleared by writing a logical one to them. Note that, unlike most other AVRs, the CBI and SBI instructions will only operate on the specified bit, and can therefore be used on registers containing such status flags. The CBI and SBI instructions work with registers 0x00 to 0x1F only.

The I/O and peripherals control registers are explained in later sections.

The ATmega4HVD/8HVD contains three General Purpose I/O Registers. These registers can be used for storing any information, and they are particularly useful for storing global variables and Status Flags. General Purpose I/O Registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

7.6 Register Description

7.6.1 EEARL – The EEPROM Address Register

Bit	7	6	5	4	3	2	1	0	
	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	X	X	X	X	X	X	X	X	

- **Bits 7:0 – EEAR7:0: EEPROM Address**

The EEPROM Address Registers – EEARL specify the EEPROM address in the 256 bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 255. The initial value of EEARL is undefined. A proper value must be written before the EEPROM may be accessed.

7.6.2 EEDR – The EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
	MSB							LSB	EEDR
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bits 7:0 – EEDR7:0: EEPROM Data**

For the EEPROM write operation, the EEDR Register contains the data to be written to the EEPROM in the address given by the EEARL Register. For the EEPROM read operation, the EEDR contains the data read out from the EEPROM at the address given by EEARL.

8.13 Register Description

8.13.1 FOSCCAL – Fast RC Oscillator Calibration Register

Bit	7	6	5	4	3	2	1	0	
	FCAL7	FCAL6	FCAL5	FCAL4	FCAL3	FCAL2	FCAL1	FCAL0	FOSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	Device Specific Calibration Value								

- **Bits 7:0 – FCAL7:0: Fast RC Oscillator Calibration Value**

The Fast RC Oscillator Calibration Register is used to trim the Fast RC Oscillator to remove process variations from the oscillator frequency. The factory-calibrated value is automatically written to this register during chip reset, giving an oscillator frequency of 8.0 MHz at 85°C. The application software can write this register to change the oscillator frequency. The oscillator can be run-time calibrated to any frequency in the range 7.3-8.1 MHz. Calibration outside that range is not guaranteed.

Note that this oscillator is used to time EEPROM and Flash write accesses, and these write times will be affected accordingly. If the EEPROM or Flash are written, do not calibrate to more than 8.1 MHz. Otherwise, the EEPROM or Flash write may fail.

The FCAL[7:5] bits determine the range of operation for the oscillator. Setting these bits to 0b000 gives the lowest frequency range, setting this bit to 0b111 gives the highest frequency range. The frequency ranges are overlapping. A setting of for instance FOSCCAL = 0x1F gives a higher frequency than FOSCCAL = 0x20.

The FCAL[4:0] bits are used to tune the frequency within the selected range. A setting of 0x00 gives the lowest frequency in that range, and a setting of 0x1F gives the highest frequency in the range. Incrementing FCAL[4:0] by 1 will give a frequency increment of less than 1.5 % in the frequency range 7.3-8.1 MHz. With an accurate time reference, an oscillator accuracy of ±1% can be achieved after calibration. The frequency will drift with temperature, so run-time calibration will be required to maintain the accuracy. Refer to ”OSI – Oscillator Sampling Interface” on page 27 for details.

8.13.2 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
	-–	–	CKOE	PUD	–	–	–	–	MCUCR
Read/Write	R	R	R/W	R/W	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 5 – CKOE: Clock Output**

When this bit is written to one, the CPU clock divided by 2 is output on the PB2 pin.

8.13.3 CLKPR – Clock Prescale Register

Bit	7	6	5	4	3	2	1	0	
	CLKPCE	–	–	–	–	–	CLKPS1	CLKPS0	CLKPR
Read/Write	R/W	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	1	1	

10. System Control and Reset

10.1 Resetting the AVR

During reset, all I/O Registers are set to their initial values, and the program starts execution from the Reset Vector. The instruction placed at the Reset Vector must be a JMP – Absolute Jump – instruction to the reset handling routine. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. The circuit diagram in Figure 10-1 on page 39 shows the reset logic. ”System and Reset Characteristics” on page 144 defines the electrical parameters of the reset circuitry.

The I/O ports of the AVR are immediately reset to their initial state when a reset source goes active. This does not require any clock source to be running.

After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the voltage regulator to reach a stable level before normal operation starts. The timeout period of the delay counter is defined by the user through the SUT Fuses. The different selections for the delay period are presented in ”Clock Sources” on page 23.

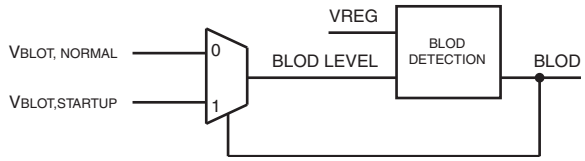
10.2 Reset Sources

The ATmega4HVD/8HVD has these reset sources:

- The Power-on Reset module generates a Power-on Reset when the Voltage Regulator starts up.
- External Reset. The MCU is reset when a low level is present on the RESET pin for longer than the minimum pulse length.
- Watchdog Reset. The MCU is reset when the Watchdog Timer period expires and the Watchdog is enabled.
- Black-out Reset. The MCU is reset when V_{REG} is below the Black-out Reset Threshold, V_{BLOT}. See “Black-out Detection” on page 40.
- debugWIRE. In On-chip Debug mode, the debugWIRE resets the MCU when giving the Reset command.

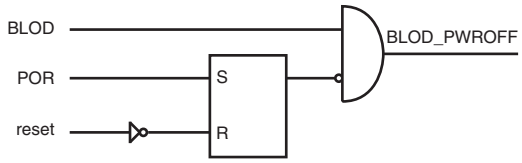
guaranteed and the chip should be forced into Power-off mode. The algorithm used for switching between the two V_{BLOT} levels is illustrated Figure 10-4 on page 41. As long as BLOD is set, the $V_{BLOT, START-UP}$ level will always be selected.

Figure 10-4. BLOD levels switching



Notice that during the Power-On Reset start-up sequence, a Black-out detection will only generate a normal reset. The chip will not enter Power-off in this case. This is illustrated in Figure 10-5 on page 41. See TBD for details on Power-on Reset and start-up sequence.

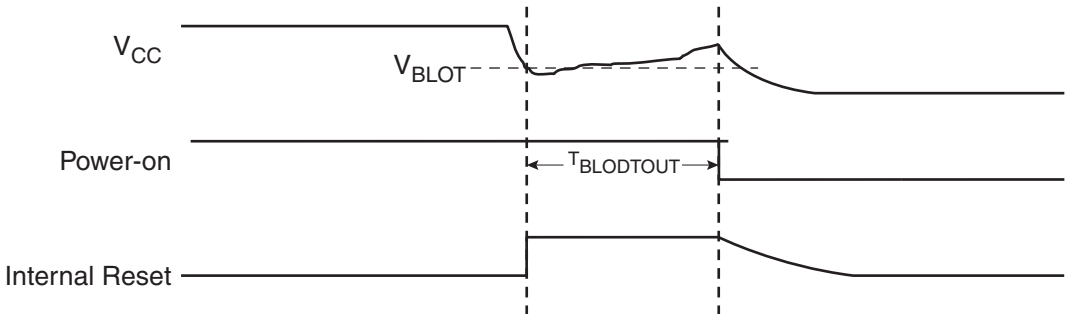
Figure 10-5. BLOD detection with POR



In normal operation, when V_{REG} decreases to a value below the trigger level, the Black-out Reset is immediately activated. After a fixed delay of $T_{BLODTOUT}$ the chip will enter Power-off mode, see Figure 10-6 on page 41 and "System and Reset Characteristics" on page 144. Any ongoing operations, including EEPROM write sequences that were started while V_{REG} was above V_{BLOD}, will be aborted. The result of an ongoing EEPROM write operation will be invalid. A charger must be connected to start up the chip from Power-off.

The BLOD circuit will only detect a drop in V_{REG} if the voltage stays below the trigger level for longer than t_{BLOD} given in "System and Reset Characteristics" on page 144.

Figure 10-6. Black-out Reset During Operation



```
Address    Labels    Code                                Comments
0x0000               rjmp       RESET                    ; Reset Handler
0x0001               rjmp       BPINT                    ; Battery Protection Interrupt Handler
0x0002               rjmp       VREGMON_INT              ; Voltage Regulator Monitor Interrupt Handler
0x0003               rjmp       EXT_INT0                 ; External Interrupt Request 0 Handler
0x0004               rjmp       EXT_INT1                 ; External Interrupt Request 1 Handler
0x0005               rjmp       WDT                      ; Watchdog Time-out Interrupt
0x0006               rjmp       TIM1_IC                  ; Timer1 Input Capture Handler
0x0007               rjmp       TIM1_COMPA               ; Timer1 Compare A Handler
0x0008               rjmp       TIM1_COMPB               ; Timer1 Compare B Handler
0x0009               rjmp       TIM1_OVF                 ; Timer1 Overflow Handler
0x000A               rjmp       TIM0_IC                  ; Timer0 Input Capture Handler
0x000B               rjmp       TIM0_COMPA               ; Timer0 Compare A Handler
0x000C               rjmp       TIM0_COMPB               ; Timer0 Compare B Handler
0x000D               rjmp       TIM0_OVF                 ; Timer0 Overflow Handler
0x000E               rjmp       ADC                      ; ADC Conversion Complete Handler
0x000F               rjmp       EE_READY                 ; EEPROM Ready
;
0x000F     RESET:    ldi        r16, high(RAMEND)        ; Main program start
0x0010               out        SPH,r16                  ; Set Stack Pointer to top of RAM
0x0010               ldi        r16, low(RAMEND)
0x0012               out        SPL,r16
0x0013               sei                                 ; Enable interrupts
0x0014               <instr>    xxx
0x0015     ...       ...        ...
;
```

14.3.1 Alternate Functions of Port B

The Port B pins with alternate functions are shown in Table 14-3.

Table 14-3. Port B Pins Alternate Functions

Port Pin	Alternate Function
PB2	MISO/CKOUT/T1 (SPI Bus Serial DataOutput, Clock Output, Timer/Counter Clock Input)
PB1	SCK/SGND/T0 (SPI Bus Master Clock input, GND for ADC0 measurements, Timer/Counter 0 Clock Input)
PB0	ADC0 (ADC Input Channel 0)

The alternate pin configuration is as follows:

- **MISO/CKOUT/T1 - Port B, Bit 2**

MISO : Slave Data Output pin for SPI programming

When not operating in programming mode, this pin can serve as Clock Output, CPU clock divided by 2. When not operating in programming mode or as clock output, the pin can be used as clock input for Time/Counter1

- **SCK/SGND/T0 - Port B, Bit 1**

SCK : Clock Input pin for SPI programming

When not operating in programming mode, this pin can serve as ground reference for ADC0 channel by configuring the pin as output low. When not used as SGND, the pin can be used as clock input for Time/Counter0.

- **ADC0 - Port B, Bit 0**

Analog to Digital Converter, channel 0.

.

Table 14-4. Overriding Signals for Alternate Functions in PB2:0

Signal Name	PB2/MISO/CKOUT/T1	PB1/SCK/SGND/T0	PB0/ADC0

16.6.2 Noise Canceler

The noise canceler improves noise immunity by using a simple digital filtering scheme. The noise canceler input is monitored over four samples, and all four must be equal for changing the output that in turn is used by the edge detector.

The noise canceler is enabled by setting the *Input Capture Noise Canceler* (ICNCn) bit in *Timer/Counter Control Register n B* (TCCRnB). When enabled the noise canceler introduces additional four system clock cycles of delay from a change applied to the input, to the update of the ICRn Register. The noise canceler uses the system clock and is therefore not affected by the prescaler.

The noise canceller should only be used for ICP01 (Port PC0).

16.6.3 Using the Input Capture Unit

The main challenge when using the Input Capture unit is to assign enough processor capacity for handling the incoming events. The time between two events is critical. If the processor has not read the captured value in the ICRn Register before the next event occurs, the ICRn will be overwritten with a new value. In this case the result of the capture will be incorrect.

When using the Input Capture interrupt, the ICRn Register should be read as early in the interrupt handler routine as possible. The maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

Measurement of an external signal duty cycle requires that the trigger edge is changed after each capture. Changing the edge sensing must be done as early as possible after the ICRn Register has been read. After a change of the edge, the Input Capture Flag (ICFn) must be cleared by software (writing a logical one to the I/O bit location). For measuring frequency only, the trigger edge change is not required.

Table 16-3. Timer/Counter0 Input Capture Source (ICS)

ICS0	Source
0	ICP00: osi_posedge pin from OSI module(1) (2)
1	ICP01: Port PC0

Notes: 1. See "OSI – Oscillator Sampling Interface" on page 27 for details.
2. The noise canceller cannot be used with this setting.

Table 16-4. Timer/Counter1 Input Capture Source (ICS)

ICS1	Source
0	ICP10: Battery Protection Interrupt(1)
1	ICP11: Voltage Regulator Interrupt(1)

Note: 1. The noise canceller cannot be used with this setting.

16.7 Output Compare Unit

The comparator continuously compares the Timer/Counter (TCNTn) with the Output Compare Registers (OCRnA and OCRnB), and whenever the Timer/Counter equals to the Output Compare Regisers, the comparator signals a match. A match will set the Output Compare Flag at the next timer clock cycle. In 8-bit mode the match can set either the Output Compare Flag OCFnA or OCFnB, but in 16-bit mode the match can set only the Output Compare Flag

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.

The Current Battery Protection (CBP) monitors the cell current by sampling the shunt resistor voltage (R_{SENSE}) connected between the NI and GND pins. A differential operational amplifier amplifies the voltage with a suitable gain. The output from the operational amplifier is compared to an accurate, programmable On-chip voltage reference by an Analog Comparator. If the shunt resistor voltage is above the Detection level for a time longer than the corresponding Protection Reaction Time, the chip activates Current Protection. A sampled system clocked by the internal ULP Oscillator is used for Short-circuit and Over-current. This ensures a reliable clock source, offset cancellation and low power consumption.

20.3 Short-circuit Protection

The Short-circuit detection is provided to enable a fast response time to very large discharge currents. If the voltage over R_{SENSE} is above the Short-circuit Detection Level for a period longer than Short-circuit Reaction Time, the Short-circuit Protection is activated.

When the Short-circuit Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the D-FET and C-FET are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the D-FET is re-enabled before the cause of the short-circuit condition is removed, the Short-circuit Protection will be activated again.

20.4 Discharge Over-current Protection

If the voltage over R_{SENSE} is above the Discharge Over-current Detection level for a time longer than Over-current Protection Reaction Time, the chip activates Discharge Over-current Protection.

When the Discharge Over-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the D-FET is re-enabled while the loading of the battery still is too large, the Discharge Over-current Protection will be activated again.

20.5 Charge Over-current Protection

If the voltage at the GND/NI pins is above the Charge Over-current Detection level for a time longer than Over-current Protection Reaction Time, the chip activates Charge Over-current Protection.

When the Charge Over-current Protection is activated, the external D-FET and C-FET are disabled and a Current Protection Timer is started. This timer ensures that the FETs are disabled for at least one second. The application software must then set the DFE and CFE bits in the FET Control and Status Register to re-enable normal operation. If the C-FET is re-enabled and the charger continues to supply too high currents, the Charge Over-current Protection will be activated again.

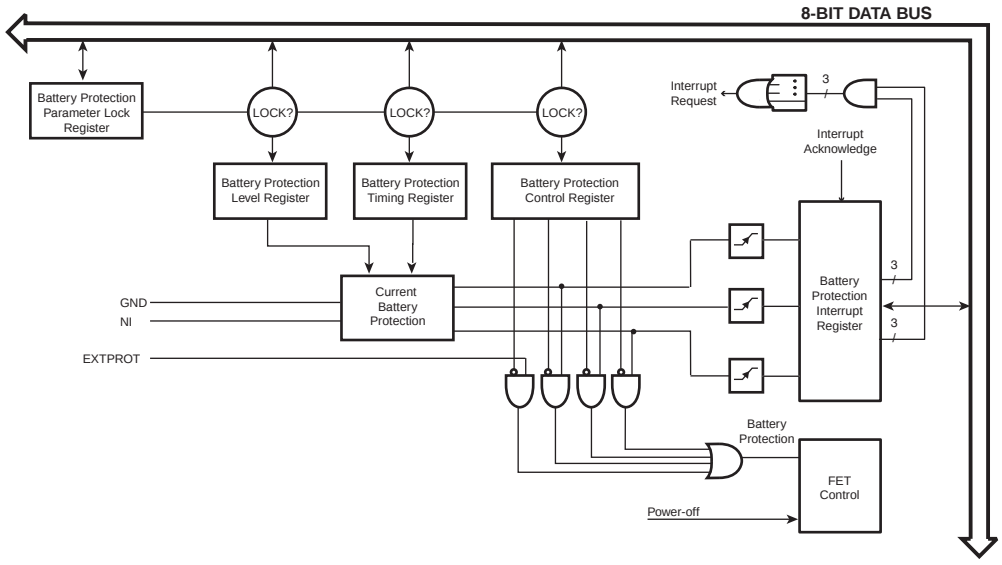
The Short-circuit and Over-current parameters are programmable to adapt to different types of batteries. The parameters are set by writing to I/O Registers. The Parameter Registers can be locked after the initial configuration, prohibiting any further updates until the next Hardware Reset.

Refer to ”Register Description” on page 108 for register descriptions.

20.7 Battery Protection CPU Interface

The Battery Protection CPU Interface is illustrated in Figure 20-2.

Figure 20-2. Battery Protection CPU Interface



Each protection originating from the Current Battery Protection module has an Interrupt Flag. All enabled flags are combined into a single battery protection interrupt request to the CPU. This interrupt can wake up the CPU from any operation mode, except Power-off. The interrupt flags are cleared by writing a logic ‘1’ to their bit locations from the CPU. An interrupt event for the External Protection Input can be generated by enabling the external interrupt for the input port.

Note that there are neither flags nor status bits indicating that the chip has entered the Power Off mode. This is because the CPU is powered down in this mode. The CPU will, however be able to detect that it came from a Power-off situation by monitoring CPU reset flags when it resumes operation.

20.8 Register Description

20.8.1 BPPLR – Battery Protection Parameter Lock Register

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	–	**BPPLE**	**BPPL**	**BPPLR**
Read/Write	R	R	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:2 – Res: Reserved Bits**

These bits are reserved and will always read as zero.

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		

- Capacitors connected to the $\overline{\text{RESET}}$ pin must be disconnected when using debugWire.
- All external reset sources must be disconnected.

22.4 Software Break Points

debugWIRE supports Program memory Break Points by the AVR Break instruction. Setting a Break Point in AVR Studio® will insert a BREAK instruction in the Program memory. The instruction replaced by the BREAK instruction will be stored. When program execution is continued, the stored instruction will be executed before continuing from the Program memory. A break can be inserted manually by putting the BREAK instruction in the program.

The Flash must be re-programmed each time a Break Point is changed. This is automatically handled by AVR Studio through the debugWIRE interface. The use of Break Points will therefore reduce the Flash Data retention. Devices used for debugging purposes should not be shipped to end customers.

22.5 Limitations of debugWIRE

The debugWIRE communication pin (dW) is physically located on the same pin as External Reset ($\overline{\text{RESET}}$). An External Reset source is therefore not supported when the debugWIRE is enabled.

The debugWIRE system accurately emulates all I/O functions when running at full speed, i.e. when the program in the CPU is running. When the CPU is stopped, care must be taken while accessing some of the I/O Registers via the debugger (AVR Studio).

A programmed DWEN Fuse enables some parts of the clock system to be running in all sleep modes. This will increase the power consumption while in sleep. Thus, the DWEN Fuse should be disabled when debugWire is not used.

22.6 Register Description

22.6.1 DWDR – debugWire Data Register

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

The DWDR Register provides a communication channel from the running program in the MCU to the debugger. This register is only accessible by the debugWIRE and can therefore not be used as a general purpose register in the normal operations.

23.5.1 EEPROM Write Prevents Writing to SPMCSR

Note that an EEPROM write operation will block all software programming to Flash. Reading the Fuses and Lock bits from software will also be prevented during the EEPROM write operation. It is recommended that the user checks the status bit (EEWE) in the EECR Register and verifies that the bit is cleared before writing to the SPMCSR Register.

23.5.2 Setting the Lock Bits from Software

To set the Lock Bits, write the desired data to R0. If bits 1..0 in R0 are cleared (zero), the corresponding Lock bit will be programmed if an SPM instruction is executed within four cycles after RFLB and SPMEN are set in SPMCSR. The Z-pointer is don't care during this operation, but for future compatibility it is recommended to load the Z-pointer with 0x0001 (same as used for reading the lOck bits). For future compatibility it is also recommended to set bit 7..2 in R0 to "1" when writing the Lock bits. When programming the Lock bits the entire Flash can be read during the operation.

Bit	7	6	5	4	3	2	1	0
R0	1	1	1	1	1	1	LB2	LB1

See Table 24-1 on page 129 and Table 24-2 on page 129 for how the different settings of the Lock bits affect the Flash access.

23.5.3 Reading the Fuse and Lock Bits from Software

It is possible to read both the Fuse and Lock bits from software. To read the Lock bits, load the Z-pointer with 0x0001 and set the RFLB and SPMEN bits in SPMCSR. When an LPM instruction is executed within three CPU cycles after the RFLB and SPMEN bits are set in SPMCSR, the value of the Lock bits will be loaded in the destination register. The RFLB and SPMEN bits will auto-clear upon completion of reading the Lock bits or if no LPM instruction is executed within three CPU cycles or no SPM instruction is executed within four CPU cycles. When RFLB and SPMEN are cleared, LPM will work as described in the Instruction set Manual.

Bit	7	6	5	4	3	2	1	0
Rd	–	–	–	–	–	–	LB2	LB1

The algorithm for reading the Fuse Low byte is similar to the one described above for reading the Lock bits. To read the Fuse Low byte, load the Z-pointer with 0x0000 and set the RFLB and SPMEN bits in SPMCSR. When an LPM instruction is executed within three cycles after the RFLB and SPMEN bits are set in the SPMCSR, the value of the Fuse Low byte (FLB) will be loaded in the destination register as shown below. Refer to Table 24-4 on page 130 for a detailed description and mapping of the Fuse Low byte.

Bit	7	6	5	4	3	2	1	0
Rd	FLB7	FLB6	FLB5	FLB4	FLB3	FLB2	FLB1	FLB0

Fuse and Lock bits that are programmed, will be read as zero. Fuse and Lock bits that are unprogrammed, will be read as one.

23.5.4 Preventing Flash Corruption

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

A Flash program corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the Flash requires a minimum voltage to operate correctly. Sec-

24.8.3 Chip Erase

The Chip Erase will erase the Flash and EEPROM[1] memories plus Lock bits. The Lock bits are not reset until the Program memory has been completely erased. The Fuse bits are not changed. A Chip Erase must be performed before the Flash and/or EEPROM are re-programmed.

Note: 1. The EEPROM memory is preserved during Chip Erase if the EESAVE Fuse is programmed.

1. Load command “Chip Erase” (see Table 24-14).
2. Wait after Instr. 3 until SDO goes high for the “Chip Erase” cycle to finish.
3. Load Command “No Operation”.

24.8.4 Programming the Flash

The Flash is organized in pages, see Table 24-10 on page 133. When programming the Flash, the program data is latched into a page buffer. This allows one page of program data to be programmed simultaneously. The following procedure describes how to program the entire Flash memory:

1. Load Command “Write Flash” (see Table 24-14).
2. Load Flash Page Buffer.
3. Load Flash High Address and Program Page. Wait after Instr. 3 until SDO goes high for the “Page Programming” cycle to finish.
4. Repeat 2 through 3 until the entire Flash is programmed or until all data has been programmed.
5. End Page Programming by Loading Command “No Operation”.

When writing or reading serial data to the ATmega4HVD/8HVD, data is clocked on the rising edge of the serial clock, see Figure 24-4, Figure 26-3 and Table 26-9 for details.

Figure 24-3. Addressing the Flash which is Organized in Pages

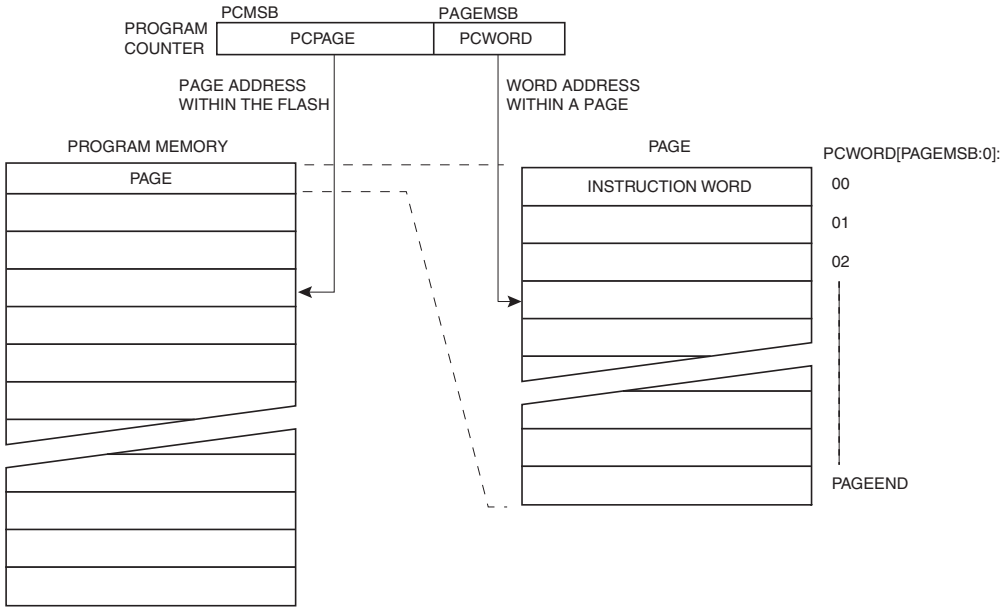


Table 24-14. High-voltage Serial Programming Instruction Set for ATmega4HVD/8HVD

Instruction		Instruction Format				Operation Remarks
		Instr.1/5	Instr.2/6	Instr.3	Instr.4	
Chip Erase	SDI SII SDO	0_1000_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx		
Load “Write Flash” Command	SDI SII SDO	0_0001_0000_00 0_0100_1100_00 x_xxxx_xxxx_xx				
Load Flash Page Buffer	SDI SII SDO	0_ **bbbb_bbbb** _00(1) 0_0000_1100_00 x_xxxx_xxxx_xx	0_**eeee_eeee**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_**dddd_dddd**_00 0_0011_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1101_00 x_xxxx_xxxx_xx	
	SDI SII SDO	0_0000_0000_00 0_0111_1100_00 x_xxxx_xxxx_xx				
Load Flash High Address and Program Page	SDI SII SDO	0_000**a_aaaa**_00 0_0001_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx		
Load “Read Flash” Command	SDI SII SDO	0_0000_0010_00 0_0100_1100_00 x_xxxx_xxxx_xx				
Read Flash Low and High Bytes	SDI SII SDO	0_**bbbb_bbbb**_00(1) 0_0000_1100_00 x_xxxx_xxxx_xx	0_000**a_aaaa**_00 0_0001_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 **q_qqqq_qqq**x_xx	
	SDI SII SDO	0_0000_0000_00 0_0111_1000_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0111_1100_00 **p_pppp_ppp**x_xx			
Load “Write EEPROM” Command	SDI SII SDO	0_0001_0001_00 0_0100_1100_00 x_xxxx_xxxx_xx				
Load EEPROM Page Buffer	SDI SII SDO	0_0**bbb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_**eeee_eeee**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1101_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx	
Program EEPROM Page	SDI SII SDO	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx			
Write EEPROM Byte(2)	SDI SII SDO	0_0**bbb_bbbb**_00 0_0000_1100_00 x_xxxx_xxxx_xx	0_**eeee_eeee**_00 0_0010_1100_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_1101_00 x_xxxx_xxxx_xx	0_0000_0000_00 0_0110_0100_00 x_xxxx_xxxx_xx	
	SDI SII SDO	0_0000_0000_00 0_0110_1100_00 x_xxxx_xxxx_xx				
Load “Read EEPROM” Command	SDI SII SDO	0_0000_0011_00 0_0100_1100_00 x_xxxx_xxxx_xx				

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.

26.6.2 High-voltage Serial Programming

Figure 26-3. High-voltage Serial Programming Timing

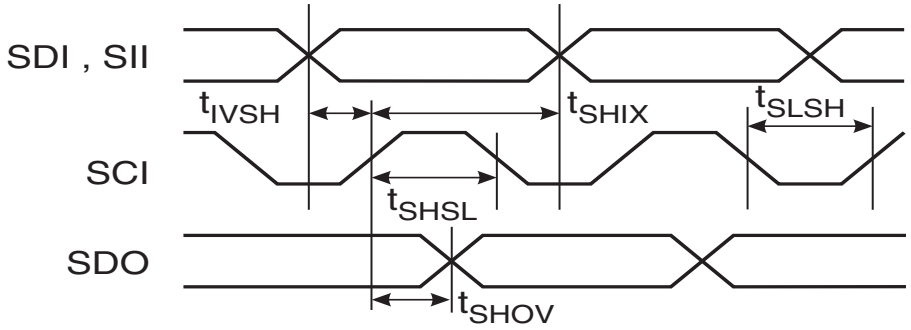


Table 26-9. High-voltage Serial Programming Characteristics T_A = 25°C ± 10%, V_{CC} = 3.5V ± 10% (Unless otherwise noted)

Symbol	Parameter	Min	Typ	Max	Units
t_{SHSL}	SCI Pulse Width High	$1/f_{ck}$			ns
t_{SLSH}	SCI Pulse Width Low	$1/f_{ck}$			ns
t_{IVSH}	SDI, SII Valid to SCI High	50			ns
t_{SHIX}	SDI, SII Hold after SCI High	50			ns
t_{SHOV}	SCI High to SDO Valid		16		ns
t_{WLWH_PFB}	Wait after Instr. 3 for Write Fuse Bits		2.5		ms

28. Register Summary

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
(0xFF)	Reserved	–	–	–	–	–	–	–	–	
(0xFE)	BPPLR	–	–	–	–	–	–	BPPLE	BPPL	
(0xFD)	BPCR	–	–	EPID	SCD	DOCD	COCD	–	–	
(0xFC)	Reserved	–	–	–	–	–	–	–	–	
(0xFB)	BPOCTR	–	–	OCTR[5:0]						
(0xFA)	BPSCTR	–	SCTR[6:0]							
(0xF9)	Reserved	–	–	–	–	–	–	–	–	
(0xF8)	Reserved	–	–	–	–	–	–	–	–	
(0xF7)	BPCOCD	COCDL[7:0]								
(0xF6)	BPDOCD	DOCDL[7:0]								
(0xF5)	BPSCD	SCDL[7:0]								
(0xF4)	Reserved	–	–	–	–	–	–	–	–	
(0xF3)	BPIFR	–	–	–	SCIF	DOCIF	COCIF	–	–	
(0xF2)	BPIMSK	–	–	–	SCIE	DOCIE	COCIE	–	–	
(0xF1)	Reserved	–	–	–	–	–	–	–	–	
(0xF0)	FCSR	–	–	–	–	DUVRD	CPS	DFE	CFE	
(0xEF)	Reserved	–	–	–	–	–	–	–	–	
(0xEE)	Reserved	–	–	–	–	–	–	–	–	
(0xED)	Reserved	–	–	–	–	–	–	–	–	
(0xEC)	Reserved	–	–	–	–	–	–	–	–	
(0xEB)	Reserved	–	–	–	–	–	–	–	–	
(0xEA)	Reserved	–	–	–	–	–	–	–	–	
(0xE9)	Reserved	–	–	–	–	–	–	–	–	
(0xE8)	Reserved	–	–	–	–	–	–	–	–	
(0xE7)	Reserved	–	–	–	–	–	–	–	–	
(0xE6)	Reserved	–	–	–	–	–	–	–	–	
(0xE5)	Reserved	–	–	–	–	–	–	–	–	
(0xE4)	Reserved	–	–	–	–	–	–	–	–	
(0xE3)	Reserved	–	–	–	–	–	–	–	–	
(0xE2)	Reserved	–	–	–	–	–	–	–	–	
(0xE1)	Reserved	–	–	–	–	–	–	–	–	
(0xE0)	Reserved	–	–	–	–	–	–	–	–	
(0xDF)	Reserved	–	–	–	–	–	–	–	–	
(0xDE)	Reserved	–	–	–	–	–	–	–	–	
(0xDD)	Reserved	–	–	–	–	–	–	–	–	
(0xDC)	Reserved	–	–	–	–	–	–	–	–	
(0xDB)	Reserved	–	–	–	–	–	–	–	–	
(0xDA)	Reserved	–	–	–	–	–	–	–	–	
(0xD9)	Reserved	–	–	–	–	–	–	–	–	
(0xD8)	Reserved	–	–	–	–	–	–	–	–	
(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	–	–	–	–	–	–	–	–	
(0xCA)	Reserved	–	–	–	–	–	–	–	–	
(0xC9)	Reserved	–	–	–	–	–	–	–	–	
(0xC8)	ROCR	ROCS	–	–	–	–	RSCDEN	RSCWIF	RSCWIE	
(0xC7)	Reserved	–	–	–	–	–	–	–	–	
(0xC6)	Reserved	–	–	–	–	–	–	–	–	
(0xC5)	Reserved	–	–	–	–	–	–	–	–	
(0xC4)	Reserved	–	–	–	–	–	–	–	–	
(0xC3)	Reserved	–	–	–	–	–	–	–	–	
(0xC2)	Reserved	–	–	–	–	–	–	–	–	
(0xC1)	Reserved	–	–	–	–	–	–	–	–	
(0xC0)	Reserved	–	–	–	–	–	–	–	–	

Mnemonics	Operands	Description	Operation	Flags	#Clocks
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC ← PC + k + 1	None	1/2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC ← PC + k + 1	None	1/2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC ← PC + k + 1	None	1/2
BIT AND BIT-TEST INSTRUCTIONS					
SBI	P,b	Set Bit in I/O Register	I/O(P,b) ← 1	None	2
CBI	P,b	Clear Bit in I/O Register	I/O(P,b) ← 0	None	2
LSL	Rd	Logical Shift Left	Rd(n+1) ← Rd(n), Rd(0) ← 0	Z,C,N,V	1
LSR	Rd	Logical Shift Right	Rd(n) ← Rd(n+1), Rd(7) ← 0	Z,C,N,V	1
ROL	Rd	Rotate Left Through Carry	Rd(0)←C,Rd(n+1)← Rd(n),C←Rd(7)	Z,C,N,V	1
ROR	Rd	Rotate Right Through Carry	Rd(7)←C,Rd(n)← Rd(n+1),C←Rd(0)	Z,C,N,V	1
ASR	Rd	Arithmetic Shift Right	Rd(n) ← Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)←Rd(7..4),Rd(7..4)←Rd(3..0)	None	1
BSET	s	Flag Set	SREG(s) ← 1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s) ← 0	SREG(s)	1
BST	Rr, b	Bit Store from Register to T	T ← Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b) ← T	None	1
SEC		Set Carry	C ← 1	C	1
CLC		Clear Carry	C ← 0	C	1
SEN		Set Negative Flag	N ← 1	N	1
CLN		Clear Negative Flag	N ← 0	N	1
SEZ		Set Zero Flag	Z ← 1	Z	1
CLZ		Clear Zero Flag	Z ← 0	Z	1
SEI		Global Interrupt Enable	I ← 1	I	1
CLI		Global Interrupt Disable	I ← 0	I	1
SES		Set Signed Test Flag	S ← 1	S	1
CLS		Clear Signed Test Flag	S ← 0	S	1
SEV		Set Twos Complement Overflow.	V ← 1	V	1
CLV		Clear Twos Complement Overflow	V ← 0	V	1
SET		Set T in SREG	T ← 1	T	1
CLT		Clear T in SREG	T ← 0	T	1
SEH		Set Half Carry Flag in SREG	H ← 1	H	1
CLH		Clear Half Carry Flag in SREG	H ← 0	H	1
DATA TRANSFER INSTRUCTIONS					
MOV	Rd, Rr	Move Between Registers	Rd ← Rr	None	1
MOVW	Rd, Rr	Copy Register Word	Rd+1:Rd ← Rr+1:Rr	None	1
LDI	Rd, K	Load Immediate	Rd ← K	None	1
LD	Rd, X	Load Indirect	Rd ← (X)	None	2
LD	Rd, X+	Load Indirect and Post-Inc.	Rd ← (X), X ← X + 1	None	2
LD	Rd, - X	Load Indirect and Pre-Dec.	X ← X - 1, Rd ← (X)	None	2
LD	Rd, Y	Load Indirect	Rd ← (Y)	None	2
LD	Rd, Y+	Load Indirect and Post-Inc.	Rd ← (Y), Y ← Y + 1	None	2
LD	Rd, - Y	Load Indirect and Pre-Dec.	Y ← Y - 1, Rd ← (Y)	None	2
LDD	Rd,Y+q	Load Indirect with Displacement	Rd ← (Y + q)	None	2
LD	Rd, Z	Load Indirect	Rd ← (Z)	None	2
LD	Rd, Z+	Load Indirect and Post-Inc.	Rd ← (Z), Z ← Z+1	None	2
LD	Rd, -Z	Load Indirect and Pre-Dec.	Z ← Z - 1, Rd ← (Z)	None	2
LDD	Rd, Z+q	Load Indirect with Displacement	Rd ← (Z + q)	None	2
LDS	Rd, k	Load Direct from SRAM	Rd ← (k)	None	2
ST	X, Rr	Store Indirect	(X) ← Rr	None	2
ST	X+, Rr	Store Indirect and Post-Inc.	(X) ← Rr, X ← X + 1	None	2
ST	- X, Rr	Store Indirect and Pre-Dec.	X ← X - 1, (X) ← Rr	None	2
ST	Y, Rr	Store Indirect	(Y) ← Rr	None	2
ST	Y+, Rr	Store Indirect and Post-Inc.	(Y) ← Rr, Y ← Y + 1	None	2
ST	- Y, Rr	Store Indirect and Pre-Dec.	Y ← Y - 1, (Y) ← Rr	None	2
STD	Y+q,Rr	Store Indirect with Displacement	(Y + q) ← Rr	None	2
ST	Z, Rr	Store Indirect	(Z) ← Rr	None	2
ST	Z+, Rr	Store Indirect and Post-Inc.	(Z) ← Rr, Z ← Z + 1	None	2
ST	-Z, Rr	Store Indirect and Pre-Dec.	Z ← Z - 1, (Z) ← Rr	None	2
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q) ← Rr	None	2
STS	k, Rr	Store Direct to SRAM	(k) ← Rr	None	2
LPM		Load Program Memory	R0 ← (Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd ← (Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Inc	Rd ← (Z), Z ← Z+1	None	3
SPM		Store Program Memory	(Z) ← R1:R0	None	-
IN	Rd, P	In Port	Rd ← P	None	1
OUT	P, Rr	Out Port	P ← Rr	None	1